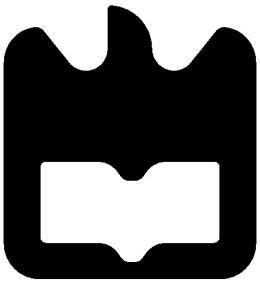
Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática,
2009

Carlos Manuel Nunes de Almeida Alves da Costa

Autenticação nos sistemas informáticos da Universidade de Aveiro



Carlos Manuel Nunes de Almeida Alves da Costa

Autenticação nos sistemas informáticos da Universidade de Aveiro

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica de Artur José Carneiro Pereira e João Manuel de Oliveira e Silva Rodrigues, Professores do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

o júri / the jury

presidente / president

Armando José Formoso de Pinho
Professor Associado com agregação da Universidade de Aveiro

vogais / examiners committee

Carlos Nuno da Cruz Ribeiro
Professor Auxiliar do Instituto Superior Técnico (arguente)

Artur José Carneiro Pereira
Professor Auxiliar da Universidade de Aveiro (orientador)

João Manuel de Oliveira e Silva Rodrigues
Professor Auxiliar da Universidade de Aveiro (co-orientador)

agradecimentos / acknowledgements

Pelas contribuições prestadas das mais diversas naturezas, não só no decorrer deste trabalho, mas também ao longo da minha vida, gostaria de agradecer aos orientadores com quem tive o privilégio de contar, aos meus colegas de trabalho, aos meus pais e à minha esposa

Resumo

A autenticação nos dias de hoje desempenha um papel, mais do que nunca, relevante na segurança da informação e dos sistemas informáticos. É com base nela que os mecanismos de controlo de acesso determinam que este último deve ou não ser concedido.

Neste trabalho descrevem-se as principais características e riscos da segurança da informação em geral e dos sistemas de autenticação, tendo em particular atenção os existentes na Universidade de Aveiro (UA). Nesta estão presentes uma grande diversidade de serviços, aplicações e sistemas informáticos cujos exigentes utilizadores (funcionários e alunos) põem à prova no dia-a-dia da vida da instituição. Há ainda que destacar a delicadeza de alguns dos primeiros que pela informação armazenada e/ou serviços disponibilizados requerem especial atenção, como é o caso dos serviços centrais da UA (DNS, DHCP, Firewall, Proxy, AD, Mail), ou aplicações/sistemas usados pelos serviços académicos e financeiros, entre outros. Uma falha, ou indisponibilidade de um deles tem nos dias de hoje grande impacto no desempenho desta instituição académica.

Foi no papel de um destes administradores que surgiu a necessidade de melhor conhecer a realidade dos sistemas de autenticação existentes hoje ao dispor dos administradores e dos utilizadores, bem como tecnologias e mecanismos. Só após esse trabalho foi possível avaliar os sistemas presentes na UA, identificando os pontos fortes e fracos de cada um e com base neles propor eventuais revisões e/ou alterações. Igualmente gratificante foi a implementação de algumas das melhorias propostas, tarefa sempre estimulante pela concretização do estudo que a precedeu.

Abstract

More than ever, authentication nowadays has a relevant role, on the security of information and of the informatics systems. It is on this base that these mechanisms of control of access determine the access.

This written work describes the main characteristics and risks of the security of the information in general and of the systems of authentication giving particular attention to those that exists at Universidade de Aveiro. At this institution there is a great diversity of services, applications and systems whose demanding users (workers and students) put those to test, every day. The fragility of some of this systems must stand out due to the information stored and/or services provided that require special attention, as it is the case of the main services (DNS, DHCP, Firewall, Proxy, AD, Mail) or applications/systems used by the academic and financial services, among others. On flaw or unavailability of one of this have nowadays a big impact on the performance of this academic institution.

By playing the role of one of these administrators there was the need of better knowing the present reality of these systems of authentication existent at the present moment and that are available to the administrators and to the users, as well as the technologies and mechanisms. Only after this work it has been possible to evaluate the systems that exist at UA, identifying the strong and the weak points of each and based on this propose some revisions and/or changes. Equally gratifying was the implementation of some of the improvements proposed, task that is rewarding by the concreteness of the study that has preceded this proposals.

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas

3DES ou 3-DES Triple DES

AD Active Directory

ADFS Active Directory Federation Services

ADN Ácido Desoxirribonucleico

AES Advanced Encryption Standard

ARCA Arquivo Central de ficheiros

AS Authentication Service

ATM Automated Teller Machine

BSOD Blue Screen of Death

CA Certication Authority

CIA Confidentiality, Integrity and Availability

CICUA Centro de Informática e Comunicações da Universidade de Aveiro

DDoS Distributed Denial of Service

DES Data Encryption Standard

DETI Departamento de Electrónica, Telecomunicações e Informática

DH Diffie-Hellman

DNS Domain Naming Service

DoS Denial of Service

DRH Direcção dos Recursos Humanos

DSA Digital Signature Algorithm

ECC Elliptic Curve Cryptography

GSSAPI Generic Security Service Application Program Interface

HTML HyperText Markup Language

IDEA International Data Encryption Algorithm

IdP Identity Provider

IPP Internet Printing Protocol

IPSec Internet Protocol Security

KDC Key Distribution Center

LDAP Lightweight Directory Access Protocol

MAC Media Access Control

MIC Message Integrity Code

MiM Man in the Middle

MIT Massachusetts Institute of Technology

NIS Network Information Service

NTLM NT Lan Manager

OASIS Organization for the Advancement of Structured Information Standards

PKI Public Key Infrastructure

QOP Quality of Protection

RC5 Rivest's Code 5

RCU Registo Central de Utilizadores

RDC Remote Desktop Client

RDP Remote Desktop Protocol

RFB Remote Framebuffer

RFC Request For Comment

RFID Radio Frequency Identification Devices

RODC Read Only Domain Controller

RPM Red Hat Package Manager

RSA Rivest, Shamir, Adleman

RSO Reduced Sign On

SACAD Serviços Académicos

SAML Security Assertion Markup Language

SIBS Sociedade Interbancária de Serviços, S.A.

SMTP Simple Mail Transfer Protocol

SP Service Provider

SRPM Source Red Hat Package Manager

SSL Secure Sockets Layer

SSO Single Sign On

TCP Transmission Control Protocol

TGT Ticket Granting Ticket

TGS Ticket Granting Service

TLS Transport Layer Security

TS Terminal Server

TSGW Terminal Service Gateway

UA Universidade de Aveiro

UDP User Datagram Protoco

UU Utilizador Universal

VIP Validação de Identificação Pessoal

VNC Virtual Network Computing

VPN Virtual Private Network

XML Extensible Markup Language

Capítulo 1

Introdução

1.1 Motivação

A autenticação nos dias de hoje desempenha um papel, mais do que nunca, relevante na segurança da informação e dos sistemas informáticos. É com base nela que os mecanismos de controlo de acesso determinam que este último deve ou não ser concedido.

Uma instituição como a Universidade de Aveiro (UA) desafia esta área da informática mais do que a generalidade das empresas ou organizações. Cerca de dezasseis mil e quinhentas pessoas estão activas no dia-a-dia do Campus de Santiago (funcionários e alunos), às quais se podem juntar mais alguns milhares de outras, que apesar de (já) não possuírem um vínculo activo com a instituição mantém um contacto próximo com a mesma. Para além do número de utilizadores há também que ter em consideração, por um lado a forma de actuar de alguns deles, por outro a interacção das múltiplas unidades que constituem a UA, e por fim, ainda a delicadeza da segurança de alguns sistemas. A isto há ainda que acrescentar a não menos grande diversidade de serviços, aplicações e sistemas informáticos existentes, cujas especificidades e âmbitos de utilização colocam igualmente reptos aos diferentes mecanismos de autenticação, bem como aos seus administradores.

Foi no papel de um destes administradores que surgiu a necessidade de melhor conhecer a realidade dos sistemas de autenticação existentes hoje ao seu dispor, bem como dos utilizadores, assim como tecnologias e mecanismos. Só após esse trabalho será possível avaliar os sistemas presentes na UA, identificando os pontos fortes e fracos de cada um e com base neles propor eventuais revisões e/ou alterações.

1.2 Objectivos

Com este trabalho pretende-se primeiro que tudo estudar os mecanismos, as tecnologias e os sistemas de autenticação existentes, mas também os usados nos sistemas informáticos da Universidade de Aveiro, com a intenção de nestes identificar deficiências, propor planos de melhoria e soluções a adoptar em aplicações ou sistemas futuros.

1.3 Metodologia

Como metodologia seguiu-se uma abordagem com ponto de partida no geral, a segurança (da informação) para a pouco e pouco se ir estreitando o campo de conhecimento até se chegar

ao caso particular dos sistemas de autenticação da UA. Pelo meio foram estudados os riscos da informação (digital), assim como mecanismos e tecnologias que permitem a detecção, e minimização do seu impacto. Em seguida foi estudada a autenticação propriamente dita: sistemas, tecnologias, mecanismos e protocolos. Chegados ao caso da UA desenvolveu-se um trabalho de análise de alguns serviços, sistemas ou aplicações, no que ao sistemas de autenticação usados diz respeito e problemas envolventes. Terminada essa análise foram indicadas algumas propostas de melhoria relativas aos sistemas de autenticação propriamente ditos, aos seus métodos, protocolos, comunicações ou mecanismos.

Como resultados espera-se obter um conhecimento dos sistemas de autenticação actualmente disponíveis, bem como dos principais sistemas da UA e encontrar um conjunto de medidas de melhoria a implementar que permitam aumentar o nível de segurança do processo de autenticação. Será também um resultado o conhecimento da problemática da "Segurança da Informação em sistemas informáticos".

1.4 Estrutura da dissertação

O presente documento é composto por cinco capítulos, dos quais esta introdução é o primeiro.

No capítulo dois são abordadas as características e os problemas da segurança da informação (digital ou não), bem como as tecnologias e os mecanismos que permitem protegê-la.

Já no terceiro capítulo estão presentes os processos de autenticação, os tipos em que se podem subdividir, assim como os protocolos mais frequentemente usados.

O capítulo quarto consiste num levantamento dos tipos de autenticações presentes da Universidade de Aveiro, bem como dos principais sistemas em que os mesmos são usados e na apresentação de propostas de revisão do sistemas identificados anteriormente de modo a que operem de forma mais segura.

O quinto e último capítulo apresenta as conclusões do trabalho, nas quais se tenta analisar e confrontar os resultados alcançados em comparação com o seu propósito, para além da indicação de alguns pertinentes trabalhos futuros.

Por fim está presente uma secção de apêndices com informação complementar à referida nos capítulos que a precedem.

Capítulo 2

Segurança

A preocupação com a Segurança de Informação é já antiga. Desde há longos anos que o Homem sente a necessidade de usar mecanismos que impeçam que mensagens secretas sejam lidas por pessoas indevidas, por nelas constar informação referente a guerras, amores, etc. É a Júlio César, Imperador Romano, que é atribuída a invenção da cifra que ganhou o seu nome, e com a qual ele pretendia enviar mensagens aos chefes dos seus exércitos, sem correr o risco de serem lidas caso caíssem nas mãos do inimigo. Apesar disto, são conhecidas anteriores formas de escrita cifrada, nomeadamente em inscrições egípcias, onde eram usados hieróglifos não padrão.

2.1 Segurança — da Informação à Informática

Por segurança de informação entende-se a protecção de informação e de sistemas de informação de serem alvo de acesso, uso, divulgação, corrupção, alteração ou destruição não autorizados. A Informação que é alvo de segurança pode ou não residir em Sistemas Informáticos. Dado que cada vez mais a informação reside nestes, é frequente serem confundidas as designações: segurança informática e segurança de informação. Apesar de estarem relacionadas e de terem alguns objectivos comuns elas possuem diferenças.

A Segurança de Informação preocupa-se essencialmente com a confidencialidade, a integridade e a disponibilidade da informação, independentemente de qual seja o seu formato: impresso, electrónico e/ou informático ou outros.

Já a Segurança Informática pode focar os seus esforços na garantia da disponibilidade e correcta operação de um Sistema Informático, sem eventualmente se preocupar com a informação nele contida ou processada.

Nos dias de hoje, a informação é recolhida, processada e armazenada em computadores e/ou sistemas informáticos. Pode pertencer às mais variadas entidades (instituições estatais, bancárias, de saúde, etc) e ser dos mais variados tipos (dados dos empregados, de fornecedores, resultados de investigações científicas, situações financeiras, etc), e flui entre sistemas, através de redes informáticas. Alguma desta informação é confidencial, por questões éticas, legais, de privacidade, rivalidade empresarial, segurança do Estado, entre outras.

Desde o final do século XX vários e rápidos avanços têm vindo a ocorrer nas áreas das telecomunicações, hardware e software informático, algoritmos de cifra, etc. A disponibilidade de cada vez mais pequenos, poderosos e baratos equipamentos informáticos tem democratizado o processamento e o armazenamento de informação, mesmo por pequenas empresas e

particulares com interligações através de redes privadas ou da Internet.

2.1.1 Confidencialidade, Integridade e Disponibilidade

Nas décadas de 80 e 90 acreditava-se que os princípios nucleares da Segurança de Informação eram a Confidencialidade (Confidentiality), a Integridade (Integrity) e a Disponibilidade (Availability). O modo como os vários componentes e agentes interagem pode ser observado na figura 2.1.

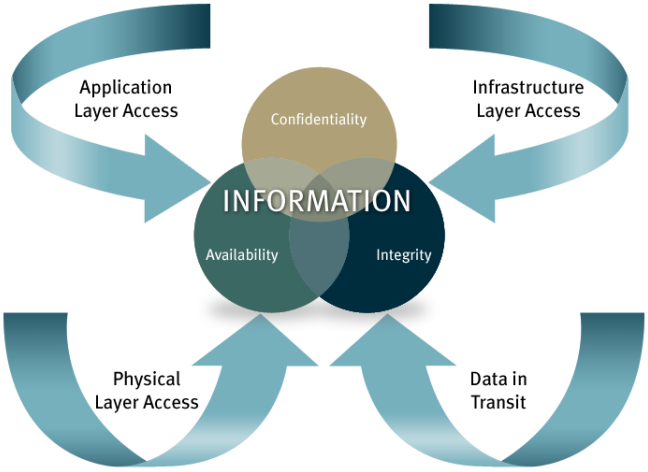


Figura 2.1: Modelo CIA[1] [1]

Confidencialidade (*Confidentiality*)

Confidencialidade é a propriedade de prevenir o acesso a informação por pessoas ou sistemas não autorizados.

Exemplo[2]: uma transacção electrónica efectuada com o recurso a um cartão de crédito implica a transmissão do número de cartão de crédito do comprador para o vendedor e deste para a rede de processamento das transacções (no caso português a SIBS).

Para que a confidencialidade se mantenha, a transmissão do número deve ser cifrada, a referência a este nos variados sistemas informáticos envolvidos deve ser a mínima e indispensável, deve ainda ser limitada às entidades intervenientes na transacção, e por sua vez o acesso aos sistemas, de forma lógica e física deve ser restrita e controlada. Se algum terceiro consegue obter um número de cartão de crédito uma falha de confidencialidade ocorreu.

A confidencialidade é necessária à manutenção da privacidade das pessoas cujos dados residem num sistema. Mas por si só não é suficiente para garantir a segurança da informação.

[1] na figura estão também representadas as camadas em que a informação digital poderá ser colocada em risco: aplicação (editores de texto, gestores de stocks, etc), infraestrutura (serviço web, base de dados, etc), comunicações (FTP, SQL, NFS, CIFS, etc) e física (bens tangíveis, electricidade, edifícios, etc)

[2] para melhor se compreender e distinguir os conceitos vamos indicar nesta e em algumas das próximas sub-secções exemplos do dia-a-dia.

Integridade (*Integrity*)

No que diz respeito à segurança de informação, a Integridade é a garantia de não alteração de informação por parte de entidades não autorizadas. Esta é violada quando alguém com más intenções apaga informação, executa um ficheiro que foi recebido por e-mail e que é portador de vírus, adultera um web site, quando engana um sistema electrónico seja ele qual seja, etc. No entanto, é também violada sem intenção, simplesmente por engano, quando alguém introduz uma informação incorrecta num sistema, por exemplo, uma aplicação mal programada adultera um campo de base de dados.

Disponibilidade (*Availability*)

Para qualquer sistema informático desempenhar o seu papel é obrigatório que o mesmo esteja disponível quando houver necessidade de ser acedido. Isto implica que os sistemas envolvidos no armazenamento e processamento da informação, nos controlos de acesso e nos canais de comunicação usados funcionem correctamente. Sistemas de alta disponibilidade pretendem conferir ao serviço essa mesma característica garantindo que, a qualquer hora, continua a operar, mesmo perante a falta de fornecimento de energia eléctrica, as falhas de hardware, as actualizações de software do sistema, os ataques, etc. No que se refere à disponibilidade de informação não é diferente. Independentemente do seu meio de suporte físico, dos variados mecanismos envolvidos, etc, o que se pretende é garantir que ela está acessível quando for necessária.

Apesar de um sistema oferecer confidencialidade, integridade e disponibilidade à informação nele armazenada, ou que nele flui, isto poderá não ser suficiente. Vamos perceber a seguir porquê.

2.1.2 Modelo de Parker

Em meados da década de 90, Donn B. Parker, um reputado especialista e pioneiro em Segurança, veio no seu livro Fighting Computer Crime[2] propor um modelo alternativo ao clássico modelo CIA: *six atomic elements of information*. Para Parker o modelo anterior era incompleto e assim acrescentou-lhe mais três propriedades/características: posse (possession), autenticidade (authenticity) e utilidade (utility), conforme consta da figura 2.2.

Autenticidade (*Authenthicity*)

Quando falamos de Segurança da Informação, em particular na área da informática e das comunicações, é imperioso garantir que a informação, as transacções, as comunicações, as identidades (digitais, ou físicas) são genuínas. É igualmente importante para se ter autenticidade, garantir que há a verificação de que as partes envolvidas em trocas de informação são efectivamente quem anunciam que são. Por exemplo, quando alguém intencionalmente forja um endereço electrónico ou de um sítio web e se faz passar por outrem, não foi quebrada a confidencialidade, integridade ou disponibilidade, mas sim a autenticidade, neste caso da identidade da pessoa ou do sítio.

Essencialmente existem dois tipos de autenticidade a garantir:

Figura 2.2: Modelo de Parker (*six atomic elements of information*) [3]

interlocutores: permite a prova ao outro interlocutor de que está a interagir com a entidade desejada e não com um impostor

conteúdos: permite ao receptor efectuar a prova de que os dados estão tal e qual foram enviados, não tendo sofrido alteração no canal de comunicação. Possibilita ainda a detecção da injecção fraudulenta de mensagens, da troca da ordem da sequência das mensagens e da repetição posterior de mensagens.

Posse ou controlo (*Possession or control*)

Quando nos referimos a posse ou controlo da informação temos um caso idêntico ao da autenticidade, ou seja, pode haver uma perda da posse sem que ocorra nenhuma falha das características do modelo CIA. Por exemplo, se uma pessoa inadvertidamente andar com cartões de crédito juntamente com o respectivo PIN, ou tiver na carteira cheques já assinados, e por azar os perder, deixou de ter controlo sobre eles e sabe que se os mesmos forem encontrados por pessoas mal intencionadas eles serão usados por estas. Apesar disto, a Confidencialidade, a Integridade e a Disponibilidade não foram violadas.

Utilidade (*Utility*)

A Utilidade é muitas vezes confundida com a disponibilidade, no entanto são características distintas, dado que a informação pode continuar disponível, mas sem ter qualquer tipo de utilidade.

Imaginemos o caso de um utilizador de um computador que opta por ter cifrada uma parte ou a totalidade dos dados armazenados em disco, e se esquece da chave que lhe permite decifrá-los. Os dados permanecerão disponíveis no entanto não terão qualquer tipo de utilidade.

Uma situação idêntica pode ser encontrada se uma cópia de segurança antiga, eventualmente em tape, persistir em bom estado, mas já não existir nenhuma *drive* operacional para aceder aos dados nela presentes. Uma quebra da utilidade da informação existirá, mas apesar disto nenhuma das restantes características da informação deixou de existir pela ausência da utilidade.

2.2 Riscos, Ameaças, Vulnerabilidades e Ataques

As características da informação (digital ou outra) descritas na secção anterior só poderão ser asseguradas se todos os riscos, ameaças, vulnerabilidades e ataques a que estão sujeitas puderem ser identificados, minimizados ou mesmo mitigados.

Para isso ser conseguido é necessário primeiro que tudo recorrer ao que vulgarmente é designado por Análise de Risco.

2.2.1 Análise de Risco

Genericamente, à palavra risco associamos algo de menos positivo, ou a alguém que o pode executar. Assim o objectivo de uma Análise de Risco é a avaliação e análise do valor dos bens, pois o valor da sua protecção é garantidamente superior ao risco de os perder. No que diz respeito à informação digital, seja quando se encontra armazenada ou a fluir entre equipamentos, quando esta é processada ou impressa, os riscos são múltiplos e de diversos tipos, pelo que todos os equipamentos que a suportam deverão ser protegidos de forma proporcional ao valor da informação mais valiosa que deles dependa.

A Análise de Risco de um bem informático (quer seja informação ou equipamento) irá tentar determinar se o seu valor é superior ao risco de ficar sem o bem. Tal como é referido em SCNA-Network Defense and Countermeasures [4], mais do que o valor tangível do bem, terá de ser avaliado o impacto de eventualmente ficar sem ele. Só com base nesta avaliação é que será possível a uma empresa criar políticas de segurança adequadas entre outras características, ao valor da sua possível perda.

A avaliação anteriormente referida consiste essencialmente em três tarefas que a seguir se descrevem sucintamente:

previsão dos riscos: Esta é talvez a mais difícil das tarefas da análise de riscos. Pode basear-se em estatísticas, ou na experiência dos técnicos que irão realizá-la. Tenta essencialmente identificar: os bens/sistemas que deverão ser alvo de protecção e segurança, as ameaças das quais se vai tentar protegê-los, e qual a probabilidade destas ocorrerem.

quantificação dos riscos: Uma vez elaborada a previsão, é possível avançar para o cálculo dos custos de protecção dos bens e/ou sistemas que são considerados como estando numa situação de risco. Para facilitar a sua identificação, as pessoas encarregues de encetar este trabalho deverão começar por aferir as consequências de não ser efectuada a protecção; posteriormente devem efectuar o levantamento dos métodos de recuperação e dos seus custos associados; e por fim, avaliar os custos de recuperação relativamente aos de protecção.

minimização ou mitigação dos riscos: Com o resultado da quantificação dos riscos é possível aos profissionais de segurança orientarem os seus esforços no sentido de diminuírem o nível de risco dos sistemas de informação da instituição/organização em questão. Para este objectivo ser conseguido é necessária a implementação de controlos de segurança para proteger os bens, sem deixar de ter em conta os custos inerentes revendo-o regularmente para serem efectuadas melhorias.

Resumidamente, a análise de riscos deve ser conduzida de modo a avaliar quais os bens que apresentam risco de falha/desaparecimento ou de falha parcial ou perda de desempenho, e ainda para quantificar o impacto de um ataque através de uma ameaça existente. Esta análise

resultará numa proposta de custo/benefício de implementação de mecanismos de protecção da informação e dos bens da instituição.

Para se poder compreender na plenitude os objectivos da Análise de Risco é necessário assimilar outros conceitos, nomeadamente os de ameaça, vulnerabilidade e ataque. Vamos abordá-los nas sub-secções seguintes.

2.2.2 Ameaças

Uma ameaça consiste numa qualquer circunstância ou evento com potencial de prejudicar uma informação, um sistema ou um bem, através de acesso não autorizado, destruição, divulgação, modificação e/ou negação de disponibilização do serviço habitualmente prestado. "(...) ameaça, é o dano que pode resultar da execução bem sucedida de um ataque."[5]

A origem de uma ameaça pode ser uma de três possíveis:

Naturais: Inundações, terramotos, furacões, deslizamentos de terra, avalanches, trovoadas, e outros eventos similares

Humanas: Eventos que são causados por seres humanos, que podem consistir em actos intencionais (por exemplo: entrada inadvertida de dados) ou acções deliberadas de rede (exemplo: fazer upload de programas maliciosos, acesso não autorizado a informações confidenciais).

Ambientais: Muitas vezes confundidas com as ameaças naturais, consistem tipicamente em falhas de energia, poluição, derramamento de substâncias químicas, fugas de gases ou líquidos, etc.

De realçar que o potencial de uma ameaça tanto pode estar presente em tentativas propositadas, ou seja, intencionais como ocasionais, isto é acidentais.

Exemplo: A Tanya ficar sem o porta-moedas/carteira, caso seja assaltada.

2.2.3 Vulnerabilidades

"Uma vulnerabilidade é uma característica de um sistema que o torna sensível a certos ataques."[5] A melhor forma de proteger alguma coisa ou alguém é conhecendo as ameaças e as suas vulnerabilidades. Só assim será possível tomar providências de maneira a minimizar os eventuais efeitos que aquelas possam causar.

Uma vez identificados os dados e os bens e sistemas expostos a riscos que se pretendem minimizar ou mesmo eliminar, é altura para avaliar as suas vulnerabilidades, que vão desde falhas de sistemas, ataques de negação de serviço, vírus, *worms*, ao administrador/utilizador do sistema que inadvertida ou propositadamente coloca em risco alguma informação ou equipamento, até mesmo à tantas vezes menosprezada componente física. Esta última pode ter origem em condições ambientais/naturais: alimentação eléctrica, cablagem de rede, etc. As vulnerabilidades podem ainda ser classificadas como internas ou externas à empresa/organização.

Exemplo: Frequentemente a Tanya anda na rua com a sua mala de mão aberta.

Controlo

Num estágio em que as vulnerabilidades se encontram bem identificadas é chegado o momento certo para se encetarem esforços no sentido de as controlar. Mecanismos como

Firewalls, *Intrusion Detection Systems*, controlos de acessos, entre outros, poderão contribuir para isso. Só recorrendo à avaliação do valor dos dados e equipamentos e dos impactos de uma eventual indisponibilidade é que se poderá identificar os mecanismos adequados a implementar. O ideal seria possuir uma rede, servidores, computadores pessoais, e demais equipamentos plenamente seguros. No entanto para além de muitas vezes este ser um cenário utópico, também implica investimentos que a empresa poderá não suportar, ou que não se justificam para, todos ou parte, dos dados/equipamentos em causa. É portanto importante identificar os bens, mas tanto ou mais, é importante classificá-los.

Classificação

Se questionarmos um utilizador sobre a importância da informação que possui armazenada no seu computador, dificilmente ele identificará um grande número de ficheiros que não tenham importância. Tipicamente aliás a resposta é "tudo, é tudo muito importante!!!". No entanto se lhe apresentarmos os custos para garantir cópias de segurança, cifragem de dados, etc, e se estes forem proporcionais à quantidade de dados ou à sua dimensão, rapidamente o utilizador começará a identificar dados que "não são assim tão importantes".

Falta então proceder à classificação da informação e dos equipamentos existentes. Vulgarmente esta passa por ser: secreta, confidencial, classificada ou não classificada. Mas pode também ser pública, ou privada, pessoal ou institucional, entre outras. Conforme se pode constatar a classificação da informação pode ter várias dimensões/categorias. Uma mesma informação pode ter várias classificações desde que respeitantes a dimensões distintas.

Em seguida, com base na catalogação realizada podem ser identificados os níveis de segurança que se deverão aplicar a cada categoria, ou combinação de categorias, em geral, passando por: alta, média, ou baixa.

2.2.4 Ataques

Em função das vulnerabilidades dos sistemas operativos, das aplicações, do hardware dos equipamentos, ou dos locais em que estes se encontram alojados poder-se-ão prever quais os ataques que poderão surgir e que deverão ser acautelados. "Um ataque é um conjunto de passos executados no âmbito da exploração de vulnerabilidades e que permitem concretizar uma acção ilícita."[5]

Exemplo: O ladrão Zézé dá deliberadamente um encontrão na Tanya de forma a que ambos caiam. O ladrão Beto Mãozinhas, que "por acaso ia por ali a passar", ajuda a Tanya a levantar-se, e simultaneamente rouba-lhe a carteira e o telemóvel.

Em seguida descrevem-se alguns dos ataques considerados mais importantes e frequentes.

Ataques à tradução de nomes

Um dos sistemas de suporte às comunicações de computadores/equipamentos de rede é o serviço de resolução/tradução de nomes (vulgo Domain Naming Service (DNS)). Basicamente este serviço consiste em disponibilizar aos seus clientes a correspondência entre nomes DNS e endereços IP, e vice-versa. Outro, mais elementar consiste na resolução de endereços MAC[3] e um IP, e vice-versa. Tipicamente esta informação existe em tabelas internas do computadores, *switchs*, *firewalls*, etc. Se um cliente efectua um pedido, e de alguma forma a resposta que

[3]Media Access Control – identificador único de cada interface de rede

obtém não é genuína, mas sim uma versão adulterada, isso poderá fazer com que o cliente passe a "falar" com um destinatário falso, julgando no entanto estar a falar com o genuíno.

Nomes enganadores Uma das formas de conseguir que um cliente direccione pedidos ou envie dados (como *logins* e *passwords*) para destinatários que ele julgue fidedignos, sem que no entanto não o sejam, é recorrendo a nomes semelhantes aos reais, ou que pareçam reais (não existindo de todo). Por exemplo, pode passar despercebido a alguns utilizadores que o sítio web para onde o *browser* será redireccionado após o *click* do rato é o www.cdg.pt e não www.cgd.pt, como ele julgará. Mais enganador será, caso o aspecto da página seja idêntico à genuína. Surgindo um formulário também semelhante a solicitar as credenciais de acesso, e sendo estas introduzidas pelo utilizador, irão passar a estar à disposição de alguém, que posteriormente as poderá usar para acesso ao sítio genuíno.

Resolução errada de DNS (*DNS Spoofing*): O *Spoofing* de DNS funciona enganando clientes de DNS de modo a redireccioná-los para outros destinos sem que estes se apercebam disto. Por exemplo um utilizador usa um *browser* e tenta aceder ao seu banco *online*. É enganado pela resolução de DNS que obtém e vai antes aceder a uma réplica do seu banco, no que diz respeito à aparência do *site*. Como não se apercebe que os sites são efectivamente diferentes, introduz como habitualmente as suas credenciais. O falso *site*, após recolher as credenciais, poderá devolver um erro do tipo estar em manutenção, de modo a que a pessoa não desconfie de nada. A partir deste momento o atacante ficou na posse das credenciais da pessoa que lhe permitem aceder ao real *site* do banco *online* e fazer com as credenciais o que bem entender.

Obtenção errada de endereços MAC (*MAC spoofing*): O *Spoofing* de endereços MAC funciona de modo semelhante ao anterior, no entanto a camada a que o ataque ocorre é mais baixa. Afecta antes a resolução de endereços MAC em endereços IP e vice-versa. Com este ataque, é redireccionado o tráfego que se destinava a um equipamento de rede/computador para outro. Este último passa assim a conseguir receber informação que numa rede comutada nunca lhe chegaria à interface de rede.

Ataques à confidencialidade

Neste tipo de ataques o objectivo final é a obtenção de dados, credenciais de acesso, ou outros, que se possam mostrar de alguma forma valiosas para o atacante. Vamos apresentar alguns exemplos.

Captura de credenciais: Caso o envio das credenciais de acesso não seja efectuada através de um canal seguro, estas poderão facilmente ser obtidas com a simples realização de captura dos pacotes IP enviados. Uma vez que as mesmas fluem sem qualquer tipo de cifra, a sua extracção está assim muito facilitada.

Tentativa de adivinhação da *password* (*Password Guessing*): Os sistemas cujo acesso depende unicamente do uso de um *login* e de uma *password* que é a generalidade dos sistemas usados na Internet e nas instituições têm um grau de segurança que depende essencialmente da dimensão e complexidade desta última. A melhor forma de garantir que

uma *password* não é facilmente comprometida é recorrendo a sequências de caractéres suficientemente complexas que evitem que a mesma possa ser identificada por quem se encontre próximo quando a mesma é inserida. Por outro lado a *password* deve ser suficientemente fácil de decorar de modo a que o utilizador não tenha de a registar e eventualmente deixar em algum sítio que um terceiro possa consultar. Algumas das formas de passwords que podem ser usadas são acrónimos das palavras de uma frase, substituição de letras por números, uso de símbolos, de modo a aumentar as combinações possíveis e a evitar os ataques mais frequentes que a seguir se descrevem. Os ataques a um sistema protegido apenas com *login* e *password* consistem essencialmente em dois: tentativa e erro (*Brute Force Password Attacks*) e ataques de dicionário. Os primeiros recorrem a algoritmos de permutações de caractéres/bytes, para testarem todas as combinações possíveis. Já os segundos tiram partido do facto da maior parte das pessoas recorrerem a palavras como simples *passwords*. Recorrendo a dicionários electrónicos tenta-se todas as palavras e/ou combinações de várias perante o sistema de autenticação em causa.

Tentativa de quebra da *password* (*Password Cracking*): Os sistemas de autenticação, por mais diversos e elaborados que sejam os protocolos implementados, guardam a informação das credenciais em ficheiros locais. Tipicamente estes encontram-se cifrados, no entanto existe um elevado número de ferramentas/programas que facilitam o *crack* das cifras, e por sua vez o acesso às *passwords*. Devem assim ser tomadas medidas que facilitem a detecção da presença de tais ferramentas, bem como o controlo do acesso físico aos servidores onde os ficheiros se encontram. Pois uma pessoa com posse física de um servidor, será uma questão de tempo até que consiga decifrar o que quer que seja.

Ataques à Autenticidade

Os ataques a esta característica estão bastante facilitados[4] no que diz respeito às comunicações baseadas no protocolo do IP, bem como TCP e UDP, isto porque nenhum deles possui qualquer tipo de mecanismo de garantia de autenticidade. É assim por exemplo permitida a falsificação dos endereços IP nos pacotes trocados *(IP spoofing)*.

Adulteração de IP (*IP Spoofing*): Basicamente um ataque do tipo *IP Spoofing* consiste na adulteração da real identidade de um sistema, fazendo-se passar por outro, e assim usufruir dos privilégios concedidos ao genuíno sistema: os pacotes enviados são modificados de modo a parecerem que são enviados com origem noutro IP (o IP atacado) passando assim a ter acesso a um servidor, a contornar um Firewall, a ser aprovado o acesso a algo controlado por *IP filtering*, etc.

Pedidos fraudulentos sobre UDP/TCP: Vários são os ataques possíveis de encetar explorando a inexistência de mecanismos no protocolo UDP/TCP que garantam a autenticidade dos agentes da comunicação. Por exemplo, quando se recorre ao protocolo NFS para acesso a uma área de informação, é vulgar que a restrição de acesso resida unicamente numa lista de IPs. Se o IP do atacante não fizer parte desta lista, e este estiver interessado em aceder à área em questão, a tarefa não é muito difícil de concretizar. Dado que o protocolo apenas verifica a

[4]no que diz respeito a IPv4 (que ainda é o protocolo usado em quase toda a Internet), porque o IPv6 já tem intrínsecos mecanismos de protecção. No entanto o seu estudo extrapola o âmbito deste trabalho

origem dos pacotes aquando da inicialização deste, ou seja, quando é efectuado o *mount*, basta aguardar que um cliente autorizado o efectue. Em seguida é-lhe apenas necessário capturar a resposta do servidor em que é fornecido o *file handler*. Daqui em diante, basta-lhe usar *IP spoofing* para aceder à informação existente na directoria montada.

Sequestro de ligações TCP (*TCP Hijack*): Este é um tipo de ataque *Man in the Middle (MiM)*. Conforme a própria designação deixa antever, consiste na presença de um atacante, ou melhor de um agente seu, entre um cliente e um servidor, sem que no entanto estes últimos se apercebam da sua presença. Muitas vezes o trabalho do agente é simplesmente ficar com uma cópia da informação trocada (dados, credenciais). Noutras situações, este altera os pacotes de modo a chegarem adulterados aos seus destinatários. No caso particular do *TCP Hijack* o cliente recebe um *reset* da sessão TCP, por parte do agente, e este, por sua vez, continua a usar a sessão previamente iniciada pelo genuíno cliente para aceder ao servidor, sem que este último se aperceba de nada.

Autoria de mensagens de correio electrónico: O protocolo que é usado no envio de mensagens de correio electrónico de um cliente para o servidor, e por sua vez na comunicação entre servidores, é o *Simple Mail Transfer Protocol (SMTP)*. Este é mais um dos muitos protocolos que não possui a obrigação de uso de qualquer tipo de mecanismo de verificação de autenticidade. Recorrendo aos campos *from* e *replyto* é possível forjar com facilidade a identidade do emissor da mensagem, bem como o destinatário das eventuais respostas.

Ataques à negação de prestação de serviços (*Denial of Service (DoS)*)

O propósito de um ataque de negação de serviço reside essencialmente na inibição do acesso e/ou utilização de um Serviço do qual utilizadores ou computadores normalmente usufruem. Não existe um único método para realizar estes ataques, mas sim uma variedade deles, em que uns exploram alguma característica ou fraqueza de protocolos ou ferramentas, outros a capacidade de processamento, a largura de banda, etc.

Exploração de vulnerabilidades conhecidas: Várias são as vulnerabilidades existentes nos equipamentos, nos sistemas operativos, nos programas, protocolos, etc. Uma vez conhecidas, estas podem ser exploradas por pessoas mal intencionadas, simplesmente para inibirem a disponibilização de determinado(s) serviço(s).

Um caso bastante fácil de implementar e que era muito eficaz (do ponto de vista do atacante, claro), era o *ping*[5] *of death*, precisamente porque era um *ping* que era mortal. Essencialmente um pacote de *ping* não pode ter mais do que 65536 bytes. No entanto, com recurso a fragmentação, tal é possível. No caso do sistema/equipamento que recebe o pacote fragmentado não estar devidamente protegido, a reconstrução do pacote pode provocar um *buffer overflow*, um *kernel panic*, um *Blue Screen of Death (BSOD)*, ou algo semelhante.

Exploração da capacidade de processamento: Neste tipo de ataques pretende-se inundar o servidor com pedidos falsos, mas bem formados, com uma quantidade tal que afecte de forma severa a resposta do mesmo, no limite, impossibilitando qualquer resposta.

[5]ping (Packet INternet Groper) - ferramenta usada para determinar se um endereço IP está ou não contactável. Para o conseguir envia um pacote e fica a aguardar que uma resposta seja emitida pelo destinatário.

Um destes ataques muito frequentes é o *syn flood*, que também é conhecido por ataque de Ack TCP. O atacante é um cliente de uma sessão TCP num servidor. Envia o pedido inicial, ao qual recebe o ACK respectivo, no entanto não responde com a aceitação do estabelecimento da sessão, mas com um novo pedido. Ao receber sucessivos pedidos, o servidor vai ficar cheio de sessões pendentes, chegando a um ponto que não consegue receber mais nenhum pedido, mesmo que seja de um cliente "bem comportado". O efeito na capacidade de processamento do servidor poderá ainda ser potenciado caso se recorra a protocolos de comunicação segura: IPSec, HTTPs, etc.

Inundação da rede: Um outro tipo de ataque de serviço é o de inundação da rede. Ao contrário dos ataques anteriores, o atacado não é directamente um servidor ou um cliente, mas sim o segmento de rede onde estes se encontram. Para isso a rede é inundada com tráfego considerado inútil, mas que consiga ter tal dimensão que a largura de banda ou a capacidade de processamento dos equipamentos de rede fiquem saturados.

Ataques distribuídos à negação de prestação de serviços (*Distributed Denial of Service (DDoS)*)

Dada a capacidade actual dos servidores, e mesmo dos equipamentos de rede (*switchs*, *routers*, *firewalls*, etc), alguns dos ataques não conseguem ser bem sucedidos se forem executados apenas a partir de uma origem. A forma encontrada de amplificação dos efeitos nefastos, passou em muito deles a estar no que se designa por ataques distribuídos. Ao contrário dos Ataques de Negação de Serviço descritos anteriormente, um ataque distribuído de DoS é iniciado em vários computadores em simultâneo. Tipicamente os clientes executam o ataque sincronizado são computadores que foram infectados por algum tipo de vírus, que aparentemente é inofensivo até ao momento em que recebe a "ordem", a hora e o alvo do ataque. À hora então comunicada, um conjunto elevado de computadores iniciam um ataque sincronizado a um servidor de modo a impedir que ele continue a disponibilizar os seus serviços habituais.

***Smurf*:** O atacado recebe uma enchente de *pings* TCP, não de uma única origem, mas sim de todos os endereços IP da própria rede. O atacante envia pedidos de *ping* para o endereço de *broadcast*, mas com o endereço de origem adulterado de modo a ter o IP do atacado. Assim todas as respostas serão enviadas para este, provocando assim um significativo impacto na capacidade de processamento dele.

***Fraggle*:** É um ataque semelhante ao *Smurf* excepto no pormenor que se baseia em *pings* UDP.

2.3 Tecnologias e Mecanismos de protecção

O acesso à informação é governado por um trinómio indissociável (ver figura 2.3): usabilidade, custo e segurança. Quando por exemplo se pretende aumentar a versatilidade de acesso, mantendo a segurança isso vai implicar um aumento do custo. Se por sua vez o aumento de custo não for possível, para se obter o desejado aumento de versatilidade irá ser necessário prescindir de alguma segurança.

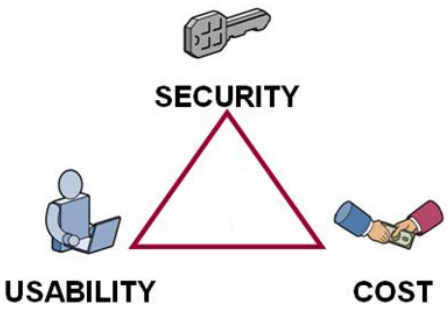


Figura 2.3: Trinónio Segurança *vs* Usabilidade *vs* Custo [6]

Quando surgiram as redes informáticas, as mesmas eram consideradas seguras, pelo menos no que a vulnerabilidades de pessoas mal intencionadas dizia respeito. Eram usadas por um número de pessoas muito limitado e controlado, e o conhecimento tecnológico e da sua existência estava ao alcance de muito poucos. Nos dias de hoje a realidade é significativamente diferente. As redes informáticas estão ao alcance de todos: já é hoje vulgar que numa empresa, cada pessoa disponha de um computador. Por outro lado, se esta empresa pretender acompanhar os desafios exponenciais que a tecnologia lhe tem vindo a apresentar (e em simultâneo proporcionar), então necessitará de proporcionar aos seu colaboradores o acesso à Internet, disponibilizar serviços a colaboradores que desenvolvam trabalho fora das instalações da mesma (comerciais, técnicos de suporte a clientes, etc), ou mesmo a clientes.

Hoje, o desafio é assim bastante grande. É necessário permitir a alguns (bem identificados) o acesso a informações/sistemas, sem que outros possam comprometer a comunicação inerente ou aproveitar-se dela de alguma forma. Essencialmente tem de ser garantido:

- canais de comunicação seguros, apesar das redes serem intrinsecamente inseguras;
- métodos de autenticação fortes, já que o tradicional *login/password* apresenta inúmeras debilidades.

De seguida descrevem-se os principais mecanismos ao dispor dos administradores de sistemas e dos especialistas de redes e segurança, que garantem uma redução da probabilidade do risco ou diminuem a probabilidade de ocorrência de falhas dos princípios da segurança de informação:

- Confidencialidade
- Integridade
- Disponibilidade
- Autenticidade
- Posse ou controlo
- Utilidade

2.3.1 Cifragem

O principal objectivo da criptografia é tornar uma informação imperceptível para quem não conheça a chave usada na sua cifragem. Essencialmente o processo consiste em cifragem e decifragem. A primeira consiste em converter a informação constante de um documento ou ficheiro em texto cru, recorrendo a uma fórmula matemática. Esta passa assim a estar cifrada. Já com a segunda é possível fazer a operação inversa: partir de informação cifrada e recorrendo a uma fórmula obter a informação decifrada, ou seja, perceptível por qualquer um, e igual à original (antes de ser cifrada).

Algoritmos

O algoritmo é o processo matemático que define como a cifragem e a decifragem podem ser encetadas. Geralmente o algoritmo é bem conhecido e não é de todo secreto (tal como não é o mecanismo de uma fechadura). O que é de todo desconhecido é a chave (ou a combinação que abre a fechadura). Ela é usada conjuntamente com o algoritmo, seja para cifrar ou decifrar informação. Logo é o componente crítico do processo, dado que qualquer pessoa pode saber como o algoritmo funciona, tal como se pode saber como funciona a fechadura.

Genericamente, quanto maior for o número de dígitos da chave, mais seguro será todo o processo e maior dificuldade terá um atacante para quebrar a chave. O ataque mais vulgar é o *brute-force* em que todas as combinações possíveis são tentadas de forma sequencial. Se a chave tiver dois ou três caracteres, e eles apenas puderem ser constituídos por dígitos de 0 a 9, então ela é possível de quebrar em pouco mais do que um segundo. Por outro lado, se puderem ser constituídos também por símbolos e letras do alfabeto e se o número de caracteres for 200 vezes maior, então o tempo será o necessário para testar todas as combinações dependendo da capacidade computacional actual. Existe a probabilidade de a chave ser quebrada à primeira tentativa... mas o tempo médio será significativamente mais elevado.

Considerando que o algoritmo não possui qualquer falha, então o alvo de um ataque só poderá residir na chave. Embora possa parecer à primeira vista estranho, esta é uma das razões pela qual o algoritmo é vulgarmente tornado público, pois assim será alvo de escrutínio por uma comunidade muito maior. Se apesar disso ninguém descobrir nenhuma falha então ele poderá ser considerado bom do ponto de vista matemático.

Os algoritmos podem-se sub-dividir em duas categorias: simétricos e assimétricos.

Algoritmos simétricos

Num algoritmo simétrico, a chave usada na cifragem é a mesma usada na decifragem. Ou seja, um emissor antes de enviar uma mensagem, cifra-a recorrendo a uma chave que conhece,

$$TextoCifrado = Cifragem(Chave, Texto).$$

O receptor ao receber a mensagem (que está cifrada), vai decifrá-la recorrendo à mesma chave, e que foi partilhada consigo anteriormente,

$$Texto = Decifragem(Chave, TextoCifrado).$$

Por exemplo: se a cifra usada fosse a de César com a rotação de apenas uma letra e a mensagem a transmitir fosse “aveiro”, então a mensagem cifrada seria “bwfjsp”. Para decifrar

Nome	Cifra	Comprimento da chave (bits)
Advanced Encryption Standard (AES)	Block Chipher	128, 192, 256
Blowfish	Block Chipher	448
Data Encryption Standard (DES)	Block Chipher	56
International Data Encry. Alg. (IDEA)	Block Chipher	128
Rivest's Code 5 (RC5)	Block Chipher	32, 64 e 128
Skipjack	Block Chipher	80
Triple DES (3DES ou 3-DES)	Block Chipher	múltiplo de 56

Tabela 2.1: Alguns dos mais comuns algoritmos simétricos de cifragem

a mensagem "bwfjsp" apenas seria necessário efectuar de novo a rotação, mas em sentido inverso, e obtinha-se a mensagem original: "aveiro".

Dado que a chave é usada quer na cifragem quer na decifragem, no caso de ela ser comprometida, toda a comunicação estará comprometida. À medida que o número de pessoas (emissoras e receptoras) envolvidas aumenta, cresce também a dificuldade de gerir e garantir que a chave não chegará a mãos indevidas, que poderão depois passar a decifrar as mensagens. Mas talvez ainda mais grave, a poder cifrá-las e enviá-las, o que fará com que sejam recebidas e aceites como tendo origem em alguém válido.

É possível identificar dois sub-tipos de algoritmos simétricos: *stream* e *block*. No primeiro a informação é cifrada *bit-a-bit*, ou *byte-a-byte*. Já na segunda ela é feita em blocos maiores, por exemplo, 64 bits. No entanto, os algoritmos mais comuns, apresentado na tabela 2.1, são do sub-tipo *block*.

Algoritmos assimétricos

A principal desvantagem dos algoritmos simétricos reside na necessidade de partilha de chave entre emissores e receptores. Já nos algoritmos assimétricos não existe esta necessidade: cada utilizador ou sistema que necessita comunicar possui a sua chave privada, secreta e intransmissível. Esta chave privada possui uma relação matemática com uma outra chave (pública), mas que apesar de ser do domínio público não permite que dela seja extraída a chave privada.

Exemplo: Quando o Carlos pretende enviar uma mensagem para a Tanya através de um meio de comunicação que não seja seguro, pode recorrer à chave pública da Tanya e cifrar a mensagem antes de a transmitir. Desta forma apesar da mensagem eventualmente passar pelas mãos de pessoas ou sistemas menos íntegros, é impossível decifrá-la sem conhecer a chave privada/secreta que apenas a Tanya conhece. Só ela a pode decifrar. Por sua vez, esta quando quiser responder à mensagem poderá efectuar o mesmo, mas desta vez recorrendo à chave pública de Carlos para cifrar a mensagem, que este decifrará recorrendo à sua chave privada.

À primeira vista os algoritmos assimétricos parecem perfeitos, e os simétricos o inverso, mas não podemos ser tão radicais na apreciação dos mesmos. Os primeiros também possuem desvantagens; por exemplo são de cálculo matemático mais exigente e portanto necessitam de maior poder computacional comparativamente com os segundos. Nenhum é perfeito, ambos têm aplicação no mundo real e a escolha depende do cenário de utilização.

Os Algoritmos assimétricos mais conhecidos são:

Diffie-Hellman (DH) é reconhecido como o "pai" dos algoritmos assimétricos. Há quem o considere em alternativa um protocolo de permuta/combinação de chaves.

Digital Signature Algorithm (DSA) A sua aplicação mais vulgar é a autenticação, mas para ser considerado seguro necessita que seja usada uma chave relativamente grande 512, ou mesmo 1024 bits. Este algoritmo foi inventado pelo *National Institute of Standards and Technology*.

Elliptic Curve Cryptography (ECC) Uma vez que recorre a elaboradas estruturas matemáticas pode ser usado em equipamentos com menores capacidades de cálculo, como telemóveis, PDA, etc.

Rivest, Shamir, Adleman (RSA) Talvez o mais vulgar de todos. É mesmo considerado como a norma dos algoritmos assimétricos.

Algoritmos mistos ou híbridos

Os algoritmos mistos ou híbridos tentam combinar as vantagens de ambos os algoritmos que os suportam (simétricos e assimétricos).

Tipicamente a solução consiste primeiro na utilização de cifra assimétrica e numa segunda fase no uso de cifra simétrica. A primeira fase serve apenas para a transferência das chaves simétricas que irão ser usadas em seguida. Posteriormente são estas as chaves usadas para a comunicação propriamente dita, ou seja, a troca de informação. Consegue-se assim aliar os aspectos positivos de cada tipo de cifra e simultaneamente colmatar os mais negativos.

Hashing

O *Hashing* tem muitas aplicações, sendo uma das mais vulgares a assinatura digital. Outra, é assegurar a integridade de determinada informação, por exemplo um download. Vulgarmente, junto com um ficheiro que podemos descarregar de um web site ou servidor FTP, está um outro pequeno ficheiro que mais não possui do que o *Hashing* do ficheiro principal. Após o descarregamento, com o recurso a uma ferramenta de cálculo de *Hashing* que use o mesmo método, poder-se-á calcular o *Hashing* do ficheiro descarregado e por fim efectuar a comparação com o valor do ficheiro anexo. Se ambos forem iguais então temos a certeza que a integridade da informação descarregada não foi comprometida.

Esta garantia pode ser dada, porque a função usada no método é de um só sentido, ou seja, não é possível partir do valor de *Hashing* e chegar à informação inicial, baseada na qual ele foi obtido. Exemplo: Vamos recorrer a um método de *Hashing* simples, que consiste em extrair a primeira letra de cada palavra de um texto. Se aplicarmos este método a um livro, e posteriormente mostrarmos o resultado do método a alguém, é virtualmente impossível a pessoa conseguir identificar o livro com base no qual o Hashing foi calculado. No entanto, já é possível que uma cópia do livro que fosse adulterada pudesse ser detectada, no caso de possuir palavras alteradas relativamente ao livro original.

Tipicamente o resultado do método é um valor fixo, ou melhor de dimensão fixa: 10, 16, 32 dígitos, etc. Quando o contexto em causa é o das assinaturas digitais, ao valor é atribuído o nome de *Message Digest*. Os métodos mais conhecidos são o MD2, o MD4 e o MD5, sendo que este último é também o de utilização mais frequente nos dias de hoje.

Se for desejável um estudo mais em detalhe podem ser consultados os seguintes livros: Cryptography and Network Security - Principles and Practice (2nd edition) [7] e Computer Security - Art and Science [8], entre outros.

2.3.2 Autenticação forte ou multi-factor

Uma boa política de *password* numa empresa, ou seja, com número de caracteres mínimo elevado, sub-conjuntos de caracteres obrigatórios (dígitos, letras do alfabeto, símbolos, etc) e prazos de expiração curtos, não é garantia de que a *password* de um utilizador não seja do conhecimento de outros utilizadores. Tal pode acontecer porque ele simplesmente a cedeu a outro, porque a *password* era fácil de adivinhar ou de alguém a memorizar ao ser vista a teclar, ou por que o utilizador possui tantos *logins* e *passwords* que os aponta num local de fácil acesso a terceiros (o método do Post'it), ou ainda porque como possui vários *logins* usa a mesma *password* em diferentes sistemas de autenticação e uma vez que alguém descubra a *password* de um sistema fica a conhecer a de todos os outros. Por fim, os sistemas em que as credenciais são usadas podem não implementar mecanismos de segurança que garantam que ninguém pode obter a *password* ao efectuar algum tipo de ataque à rede ou a sistemas.

Com isto pode-se concluir que com o método de autenticação com recurso apenas ao *login* e *password* não há a garantia de que efectivamente quem está a apresentar as credencias é o dono das mesmas. Uma resposta a esta debilidade reside no uso de autenticação forte.

Por autenticação forte entende-se uma autenticação que é mais segura no que à prova da identidade da pessoa diz respeito. Isso é possível usando em simultâneo vários factores, por exemplo, o que a pessoa sabe (*login/password*) e o que a pessoa tem (um cartão). No caso do acesso à informação por pessoas não autorizadas possuir um custo já calculado e que seja elevado o suficiente, poder-se-á mesmo recorrer não a dois factores, mas a três. Consegue-se assim obter uma maior garantia da identidade de quem está a aceder ao sistema em causa.

Quando um sistema é crítico, ou a informação deste a que se está a aceder o é, o recurso a uma autenticação multi-factor é mais do que obrigatória.

Uma das situações em que é imprescindível a autenticação forte é quando se pretende garantir o princípio da não repudiação (*non-repudiation*), por exemplo, nas transacções comerciais electrónicas. Esta consiste em se conseguir provar de que uma comunicação existiu: quer porque quem enviou consegue provar que foi o próprio que enviou, quer porque também não consegue negar que o fez, o mesmo acontecendo para o receptor.

Os factores que podem ser usados são os seguintes:

Conhecimento: algo que se sabe (ex: password/PIN)

Posse: algo que se tem (ex: um cartão, um certificado)

Físico/Biométrico: algo que se é (ex: impressões digitais, voz, olhos)

O primeiro tipo de factores é o mais vulgar. Quem se pretende autenticar tem de fornecer algo que conhece: um PIN, uma *password*. Infelizmente é algo que outros podem conhecer ou tentar adivinhar.

O segundo tipo também é comum, mas não em todos os formatos que podem existir. O mais frequente de ver é o cartão de multibanco ou de crédito. Mas existem outros, nomeadamente cartões RF-ID, ou mesmo *tokens* e *smart-cards* que são menos conhecidos do utilizador comum.

Quando uma pessoa se dirige a uma caixa de multibanco ou terminal de pagamento automático (ATM), está sem se aperceber a efectuar uma autenticação de dois factores: o cartão que possui e o PIN que conhece. Sem ambos não é possível efectuar qualquer tipo de pagamento, movimento ou mesmo consulta da conta. Caso assim não fosse e bastasse fornecer um nome e um PIN, ou apresentar unicamente um cartão, então as instituições financeiras teriam de lidar nos seus balcões com muitos casos de utilização fraudulenta e garantidamente que os seus clientes rapidamente abandonariam a utilização destes serviços.

O terceiro tipo de factores está pouco a pouco a ganhar presença no dia-a-dia de alguns utilizadores, cuja prova de identidade se requer mais exigente. Características biométricas como particularidades dos olhos (retina ou íris), da voz (timbre, tonalidade), dos dedos, são difíceis senão mesmo impossíveis de forjar. Pela garantia superior que dão é de prever que a presença deste factor venha a aumentar significativamente nos próximos anos. Há que referir que com estes mecanismos não é o utilizador que tenta provador que é a pessoa A ou B, mas é o sistema que tem de conseguir reconhecer que por exemplo perante o leitor de íris se encontra o utilizador A.

É possível combinar quaisquer dos sistemas anteriores na autenticação do acesso a sistemas ou informação.

Exemplo: Imaginemos um cofre forte alugado pelo Sr. Fernando, que para se aceder é necessário mostrar o Bilhete de identidade ao segurança para entrar no recinto, posteriormente introduzir um par *login/password* para abertura da porta do edifício, em seguida passar um cartão com PIN associado para acesso ao elevador, e ainda uma sequência de portas de corredor que são abertas com recurso a mecanismos biométricos, íris e voz e por fim a porta do cofre que é aberta com impressão digital. Será com certeza muito difícil de imaginar a possibilidade de alguém que não o Sr. Fernando venha a conseguir aceder ao conteúdo do cofre.

Podemos assim concluir que se queremos ter sistemas e informações neles contidas efectivamente seguras então é obrigatório recorrer para além do primeiro factor, também a mecanismos do sub-conjunto do segundo e terceiro factores.

2.3.3 Autenticação mútua ou bidireccional

A autenticação mútua ou bidireccional existe quando ambas as partes de uma comunicação conseguem provar a sua identidade uma à outra. Tipicamente o servidor precisa de confirmar o utilizador que pretende aceder a um determinado recurso. Mas por sua vez, o utilizador pode querer que seja garantido que o servidor a quem ele está a solicitar acesso a um recurso, e a quem irá entregar as suas credenciais para provar a sua identidade, é efectivamente quem supostamente diz ser.

Obviamente, além de vantagens também possui desvantagens. Duas entidades, para poderem comunicar usufruindo de autenticação mútua baseada em certificados digitais vão necessitar de, previamente à comunicação, trocarem os seus certificados e que estes sejam aceites mutuamente. Imaginemos agora o caso em que não são apenas duas entidades, mas dezenas ou mesmo centenas de entidades envolvidas. Isso implica que seja efectuada uma gestão bastante complexa, para que se consiga garantir que todas as entidades possuem os certificados de todas as restantes com quem eventualmente poderão comunicar. Só assim será possível que cada uma das entidades envolvidas numa comunicação valide a identidade da outra.

Um dos protocolos que implementa autenticação mútua é o Kerberos, sendo talvez uma

das razões por que é tão popular. O método recorre ao uso de chaves de cifragem baseadas em *passwords*, seja de utilizadores, seja de máquinas e que supostamente são apenas do conhecimentos dos próprios e do servidor de autenticação.

Capítulo 3

Autenticação

No decurso deste capítulo vamos ver com mais detalhe como funciona o mecanismo de autenticação, em que informação se pode ele basear para tomar a decisão de validar ou não uma autenticação, quais são os principais tipos de autenticação e ainda quais os outros mecanismos que existem e com os quais o comum dos utilizadores a confunde. Identificação e Autorização são os casos mais vulgares. Vamos assim tentar perceber o que os caracteriza, e que também os diferencia.

3.1 Autenticação – o mecanismo

Autenticação é o processo de determinar se um utilizador ou uma identidade electrónica é de quem diz que é. A autenticação é conseguida utilizando alguma coisa:

- que o utilizador sabe (ex: *password*)
- que o utilizador tem (ex: *security token*)
- do utilizador (ex: dado biométrico)

Este processo pressupõe uma avaliação do risco. Um sistema, aplicação ou informação de elevado risco ou importância requerem formas de autenticação que ofereçam maiores garantias, do que outras de baixo risco, onde a confirmação da identidade digital não é tão valiosa, no que ao risco diz respeito. A primeira forma de autenticação é vulgarmente designada por autenticação forte.

Os processos de autenticação são baseados na verificação da identidade e no seu registo.

Exemplo: Quando um novo funcionário é admitido numa empresa é-lhe solicitado que forneça determinadas informações e que apresente cartões e/ou certidões que comprovem informações (nome, morada, data de nascimento, número de bilhete de identidade e de contribuinte, carta de condução, etc). Nesse momento a empresa pode simplesmente aceitar esta informação, ou pode despoletar um processo de verificação das informações. Se estas forem verdadeiras, a empresa aceitará a sua identidade e registá-la-á nos sistemas informáticos. Aquando deste registo do novo funcionário são gerados/emitidos normalmente os mecanismos de autenticação usados na empresa: *login*, *password*, *security token*, certificados digitais e/ou registados alguns dados biométricos.

Todos os processos de autenticação que decorrerão são baseados na validação da identidade, que por sua vez dependem do processo registo do funcionário. Se este tiver fornecido

cartões/certidões falsas e que tenham sido aceites pela empresa, a pessoa que se está efectivamente a fazer passar por outra será autenticada positivamente em todos os processos de autenticação, apesar de não ser de facto quem reivindica ser. Com isto se conclui que o processo de autenticação é tão bom quanto for o mais fraco de todos os elementos da cadeia de processos de identificação, verificação, registo, etc.

3.2 Autenticação *vs* Identificação

A autenticação é conseguida através de um processo ou um sistema em que o utilizador prova determinada informação (tipicamente quem ele é).

Exemplo: Quando uma pessoa assina um documento, a autenticidade da assinatura é verificável através da sua comparação, com a de um cartão de identidade. Ou seja, o sistema usa a afirmação proferida pelo utilizador (que é quem possui a assinatura) e vai verificar se coincide com a informação existente sobre a mesma, na base de dados (cartão de identidade). É assim realizada uma correspondência de um para um.

Já num sistema/processo de identificação o próprio sistema é que vai tentar identificar ou reconhecer a pessoa que está perante ele. Tipicamente uma pessoa reconhece outra com base em aspectos ou características físicas: cor de cabelo, altura, forma facial, etc. Neste caso, o utilizador não afirma que é a pessoa com a identidade A ou B. Vai sim ser o sistema a pesquisar toda a base de dados pela coincidência das características e com base nela obter a pessoa a quem pertencem. Aqui temos uma correspondência de muitos para um.

Resumidamente, a identificação é o processo que determina a identidade de um objecto, tipicamente de um indivíduo, enquanto a autenticação é o processo de determinar se o indivíduo consegue provar determina informação. Na primeira existe unicidade, na segunda já não.

No âmbito deste documento, salvo onde for realizada a devida ressalva, a utilização da palavra identificação deverá ser interpretada como autenticação.

3.3 Autenticação – Tipos e variantes

Autenticação baseada em *password*

O método mais comum de autenticação é o que se baseia no conhecimento de uma *password*. Apesar disto é este o menos seguro de todos. A definição política do comprimento, tipo de caracteres, duração e histórico das *passwords* a serem usadas não é uma tarefa fácil. O crescimento do poder computacional tem vindo a permitir que a quebra de *passwords* ocorra cada vez com mais facilidade. Nos casos em que se justifique, para combater esta falha de segurança recorre-se a uma estratificação em vários níveis de segurança. À medida que se vai acedendo a níveis de informação/sistema/aplicação de risco mais elevado vai-se também requisitando ao utilizador que vá fornecendo autenticações mais robustas.

Autenticação recorrendo a LDAP

A grande maioria das empresas recorre a repositórios de Lightweight Directory Access Protocol (LDAP) de forma a centralizar a sua informação, nomeadamente as credenciais dos seus funcionários. Exemplos de LDAPs são o Active Directory (AD), o *Sun One Directory*, o *Novel e-Directory*, entre outros. Com estes conseguem-se realizar autenticações e pesquisas de

identidade de forma mais fácil e rápida do que recorrendo às tradicionais bases de dados, o que é particularmente importante em instituições/organizações de grande dimensão. Actualmente é ainda possível a constituição de repositórios virtuais, que mais não fazem do que integrar num só as identidades e informações de autenticação existentes em vários LDAPs e bases de dados. Com este tipo de infraestrutura de identidade é também possível integrar o controlo de acessos.

Autenticação biométrica

O processo de autenticação perante um repositório de identidades, ou uma base de dados, recorrendo a um ou mais aspectos físicos, partes ou características digitalizadas destes, designa-se por autenticação biométrica. Os tipos mais vulgares são impressões digitais, de mãos, de retinas de olhos, da assinatura, reconhecimento de voz, etc. Começa também a tornar-se mais comum o recurso a ADN.

Autenticação baseada em certificados digitais

Uma outra forma de autenticar um utilizador é recorrendo a uma *Public Key Infrastructure (PKI)*. Para cada utilizador é atribuído por uma Autoridade Certificadora (*Certification Authority - CA*) um certificado digital. Este será apresentado aquando de um processo de autenticação de forma a confirmar que a identidade é efectivamente de quem pressupostamente se diz. O grau de confiança da autenticação neste caso dependerá do nível de verificação da identidade que foi efectuada aquando do processo de registo, bem como do processo de revogação dos certificados. Este tipo de autenticação tem vindo a tornar-se cada vez mais vulgar em sistemas de SSO, de gestão documental e em sítios web.

Autenticação com *Security Tokens*

Os *tokens RSA secureID* são o mais vulgar dos *Security Tokens*. Estes são usados para autenticar uma identidade através de algo que se possui. Durante um processo de *login* ou a passagem a uma área de maior risco de informação ou aplicação é solicitado ao utilizador que introduza os dígitos do número que surge no mostrador do dispositivo que possui. Este número mais não é do que uma *password* que é gerada pelo dispositivo baseada num *PIN code* que o utilizador sabe (e na hora em que o processo está a ocorrer). Como a *password* é diferente apesar do *PIN code* ser o mesmo, a autenticação é mais segura do que a simples introdução de uma *password* que não muda durante o período de tempo definido pela política de *passwords* da empresa. Outro tipo de *security tokens* é o *Smart card*. A sua utilização tem tido um crescimento significativo nos últimos anos. Por vezes neles está contida informação variada sobre a pessoa a quem pertence (nome, número de identificação morada, e-mail, etc) e/ou certificados digitais. Algum ou mesmo todo o conteúdo contido no cartão só é acessível através de um PIN. Mais uma vez temos aqui dois factores de autenticação: algo que se sabe e algo que se possui. Uma vantagem adicional de um cartão destes é que pode incorporar num só dispositivo físico vários tipos de suporte de informação: *chip*, *RFID*, banda magnética, etc. Um exemplo deste cartão é o Cartão de Cidadão.

Autenticação forte

Quando se usa o termo autenticação forte, pretende-se transmitir a ideia de que a mesma possui um nível de confiança mais elevado que o habitual. Como métodos de autenticação forte estão incluídos os certificados digitais, *tokens* de segurança, dados biométricos, etc. Para aumentar ainda mais a confiança de uma autenticação é possível o recurso à utilização combinada de vários métodos de autenticação forte.

Autenticação com recurso a *Single Sign On (SSO)*

O *Single Sign On* consiste essencialmente na simplificação da utilização dos recursos por parte dos utilizadores, através da eliminação da necessidade de (re)autenticação de um utilizador quando este acede a um recurso após já se ter autenticado anteriormente no acesso a outro. Este mecanismo vem indirectamente permitir que os administradores implementem melhores regras relativas à segurança das *passwords*, nomeadamente no que diz respeito ao comprimento e à complexidade das mesmas, sem prejuízo significativo para o utilizador, pois a necessidade de as introduzir diminui drasticamente.

Autenticação Federada

A possibilidade de confiar numa identidade electrónica que é apresentada perante uma empresa, mas que é proveniente de outra, sua parceira, é designada de autenticação federada ou federativa. Diversos protocolos vieram permitir este tipo de autenticação; os exemplos mais vulgares são: *Security Assertion Markup Language (SAML)*[1], *Liberty Alliance*, *Web Services Federation* e *Shibboleth*. Quando usada conjuntamente com SSO, o utilizador vê a sua vida ainda mais facilitada, uma vez que não irá necessitar de se lembrar de mais um par de identificador e *password* para aceder a aplicações, sítios Web, ou outros recursos de instituição terceira. Com isto consegue-se também que a política de *password* seja a mesma, que a já usada internamente na instituição. Este tipo de autenticação ao ser implementado vem também anular toda a componente de gestão de identidades, verificação de credenciais apresentadas, para os utilizadores de instituições com as quais haja uma relação de confiança federativa estabelecida. Ou seja as vantagens verificam-se em ambos os lados, quando falamos de acesso a recursos/serviços partilhados entre instituições diferentes.

3.4 Autorização – Controlo de Acessos

Ao processo de atribuir ou não o acesso de uma pessoa a uma zona de um edifício, à utilização de um equipamento, ou algo semelhante, designa-se por controlo de acessos. Este pode ser conseguido das mais variadas formas. Pode consistir na presença de uma pessoa, vulgarmente designada por segurança, que após verificação da identidade da pessoa que pretende aceder, apresentando esta algum documento com fotografia, decide se é ou não permitida a passagem. Um outro caso presente no dia-a-dia de todos nós é o uso da campainha, acompanhada de um interfone, ou mesmo de videofone; após a identificação da

[1]SAML é uma norma do Security Services Technical Committee da Organization for the Advancement of Structured Information Standards (OASIS) que tem por base Extensible Markup Language (XML) e que permite a troca de informação relativa a autenticações e autorizações.

identidade da(s) pessoa(s) então poderá ser decidido se é ou não concedida autorização para entrar.

Com o acesso aos recursos electrónicos, quer sejam os que estão presentes no computador em que o utilizador se encontra, quer noutro computador ou sistema remoto que esteja acessível via rede, acontece algo de semelhante. A identidade electrónica do utilizador terá de ser verificada, ou seja, terá que ocorrer uma autenticação e posteriormente a esta irá então ocorrer (ou não) a autorização de acesso ao recurso que o utilizador pretende utilizar.

A decisão de concessão do acesso ao recurso pode passar por algo tão simples como quem está autenticado pode imprimir. Por outro lado, pode ser algo um pouco mais elaborado, ou complexo. Por exemplo, para utilização das impressoras de um gabinete de arquitectura, as regras podem ser:

- imprimir para a plotter, com recurso à tecnologia de cor – todos os projectistas da empresa que pertençam ao grupo de trabalho do Projecto RIA
- imprimir para a plotter, com recurso à tecnologia monocromática – todos os projectistas da empresa
- imprimir para a impressora laser monocromática – todos os funcionários

Neste gabinete, quando um utilizador pretender imprimir na impressora laser, terá apenas de ser garantido que o mesmo se encontra previamente autenticado, ou tê-lo-á de fazer no momento em que pretende usar a impressora. Já no caso de um utilizador que pretenda imprimir um desenho, a cores, com recurso à plotter, para além da autenticação, terá também de ser verificado que ele é projectista e que está envolvido no Projecto RIA. Para isso, por exemplo, no sistema de LDAP poderão existir grupos que representem nomeadamente os dois grupos (projectistas e elementos do projecto RIA) e o servidor de impressão irá consultá-los para confirmar se o utilizador pertence ou não simultaneamente aos dois grupos e, só se pertencer, será autorizada e processada a impressão.

3.4.1 Autenticação *vs* autorização

É vulgarmente feita confusão entre autenticação e autorização, pela sua estreita ligação, bem como por alguns dos sistemas que implementam a primeira também implementar a segunda.

Exemplo: Um portador de um cartão de RFID usa-o junto de um leitor, este faz a leitura do número do ID e envia-o para uma unidade de controlo. Em seguida, esta consulta a sua base de dados e confirma se o número está registado. Se estiver, vai ainda ser verificado se para este número está concedido acesso à zona protegida pela porta, junto à qual se encontra o leitor. Em caso afirmativo a porta será destrancada.

Neste exemplo, a autenticação consiste, única e exclusivamente na verificação da identidade do cartão e por inerência do suposto portador. Já a autorização é encetada apenas no caso da autenticação ser positiva e para determinar se deve ou não ser concedido o acesso a determinada zona, através da abertura ou não da porta. Na realidade, quase que podemos considerar ambos os passos como um único.

3.4.2 Auditoria

Os sistemas de autenticação e/ou de autorização também implementam vulgarmente o registo das verificações por eles efectuadas – *Auditing*. Este registo pode ser muito elementar

(ex: *login*, sucesso/falha), ou algo mais completo (para além das anteriores: data, hora, IP e porto de origem, protocolo usado, etc). Com ele poderão ser efectuados diagnósticos de falhas de autenticação ou acesso a recursos, geradas estatísticas e efectuadas outras análises sobre os sistemas. Por outro lado com algoritmos inteligentes poder-se-ão tentar detectar tentativas de quebra de *passwords* de um utilizador ou de vários.

Exemplo: No caso de haver dez tentativas falhadas de autenticação em menos de trinta segundos poder-se-á considerar que não pode ser um utilizador que possa estar a tentar autenticar-se e com mais certeza ainda se poderá deduzir que não é o dono da conta de utilizador que a estará a tentar (se fosse não falharia tantas vezes, em tão pouco tempo), pelo que poderá ser enviado automaticamente um e-mail aos administradores dos sistemas de autenticação a alertar para o facto.

3.5 Protocolos de autenticação (e/ou autorização)

Até agora temos vindo a identificar características da informação, mecanismos de autenticação e algumas propriedades destes, sempre numa perspectiva generalista. É chegada agora a altura de procurar nos protocolos existentes, e que se encontram à disposição dos programadores bem como dos administradores de sistemas, estas mesmas características e propriedades. Quais deles as possuem, em que casos eles se utilizam, o que os diferencia e qual a sua abrangência, são algumas das perguntas que vamos tentar responder na presente secção.

3.5.1 Protocolos – problemas, limitações e vantagens

Quando um administrador de sistemas tem de optar por um sistema de autenticação (ou de autorização), primeiro que tudo tem de enumerar as características que necessita que eles possuam para perante os disponíveis poder efectuar a escolha acertada para a empresa/instituição em causa. Nem sempre o que é melhor para a Empresa A, significa que também o será para a Empresa B. Tipicamente no conjunto de características procuradas estão a abrangência, a possibilidade de recorrer a autenticação forte, autenticação mútua, tipos de criptografia usados, entre outros. Vamos então ver mais em pormenor, na tabela 3.1, os protocolos que são mais vulgares de encontrar implementados, bem como o que os diferencia.

Conforme é possível observar os protocolos que são simultaneamente mais versáteis e completos são o LDAP e o Kerberos. Ambos são independentes da plataforma (Windows, Linux/Unix, etc), possuem uma norma bem definida (RFCs) e, por coincidência ou não, foram inventados em universidades/institutos de investigação e não por fabricantes.

O protocolo Kerberos possui cifra nativa, enquanto o LDAP apenas a possui como opcional. No que diz respeito à *password* de um utilizador que pretende ser autenticado ou aceder a um recurso, no caso do Kerberos ela nunca irá fluir na rede; o contrário acontece no LDAP – mais uma razão para ser imprescindível o recurso ao SSL – pois irá fluir em texto simples.

O LDAP possui a vantagem de ser um sistema cujo principal propósito é o de ser um repositório de informação de utilizadores (Directório), pelo que, com base nele, é possível autorizar ou não o acesso a recursos. Dado que o Kerberos é apenas um protocolo de autenticação, ele por si só não consegue implementar autorização. Por esta razão estes protocolos surgem frequentemente em conjunto sendo assim explorados os pontos fortes de cada um: o

Protocolo **Propriedade**	**LDAP** (Light Directory Access Protocol)	**NTLM** (NT LAN Manager)	**NIS** (Network Information Service)	**Kerberos**
Origem / invenção	University of Michigan	Microsoft	Sun Microsystems	Massachusetts Institute of Technology (MIT)
Standard	RFC 4511, 4513, independente da platforma	protocolo de autenticação desenvolvido pela Microsoft	dependente da plataforma	aberto, baseado no RFC 4120
Autorização	sim	sim	sim	standard não, extensão MS sim
Autenticação forte	nativamente não, apenas através do GSSAPI	não	não	sim
Autenticação mútua	não	não	não	opcionalmente pode ser usado caso o cliente suporte
Criptografia	SSL opcional	v1: RFC 2433; v2: RFC 2759	Só no caso do NIS+	simétrica: DES de base; podem ser usadas outras
password	circula do cliente para o servidor aplicacional; deste para o servidor de LDAP	sim, cifrada	sim	nunca circula na rede
Delegação	não	não	não	suporta vários tipos: delegação total; delegação parcial (*Constrained Delegation*); e delegação parcial com transição de protocolo (versão 2003 ou mais recentes)
Dependências	não	não	NTP	NTP e DNS

Tabela 3.1: Protocolos de autenticação mais vulgares

Kerberos faz a autenticação e o LDAP a autorização. Este processo conjunto pode funcionar de duas formas:

- o utilizador apresenta as credenciais Kerberos junto do servidor que possui o recurso ao qual ele pretende aceder; este, após confirmar a veracidade das credenciais, vai obter, via LDAP, as informações sobre o utilizador para determinar se concede ou não o acesso ao recurso

- o servidor ao detectar um utilizador a tentar aceder a um recurso, e não estando o primeiro ainda autenticado vai obrigá-lo a fazê-lo perante o LDAP, mas em vez de recorrer ao mecanismo base do protocolo, vai delegar esta tarefa a uma interface aplicacional externa, *Generic Security Services Application Programming Interface (GSS-API)*. Esta *framework* permite que as aplicações que a usem possam, entre outras coisas, vir a evoluir no que diz respeito à componente de autenticação, sem que a aplicação propriamente dita tenha de ser re-escrita. Como um dos mecanismos a que o GSS-API pode recorrer é o Kerberos, isto possibilita que o LDAP possa ter clientes cuja autenticação é efectuada com recurso a este protocolo. Após a autenticação, o processo de autorização decorre de forma idêntica ao caso anterior.

Apesar do Kerberos de raiz não implementar autorização, através da exploração de um campo que neste se encontra disponível, a Microsoft decidiu implementá-la com base na informação existente no repositório LDAP que faz coexistir num só servidor. A esta implementação conjunta do serviço de Kerberos e do serviço de LDAP o fabricante dá o nome de *Active Directory* e por sua vez a cada servidor *Domain Controler - DC*.

O protocolo Kerberos tem vindo a evoluir e por sua vez a implementar novos mecanismos, nomeadamente de delegação de poderes. Por exemplo, no caso de um servidor web que tenha sido autorizado a usar a delegação, quando um utilizador se autenticar perante ele, será possível ao servidor obter credenciais para se apresentar junto de um outro servidor como se do próprio utilizador se tratasse,(ex: base de dados), e efectuar nele operações usufruindo das permissões atribuídas ao utilizador em causa e não de permissões genéricas, atribuídas ao servidor ou ao serviço em causa. Por uma questão de segurança esta delegação poderá ser parcial, ou seja, pode estar limitada a algum ou alguns serviços que estejam em execução no servidor e não à sua totalidade.

3.6 Kerberos

Cerberus, na mitologia grega é o nome do cão de três cabeças que guarda a entrada para o reino das trevas. A designação inglesa atribuída a este sistema é Kerberos porque é constituído essencialmente por três partes:

- servidor de autenticação *Key Distribution Center (KDC)*, cujo âmbito é designado por *REALM* (reino);

- utilizador, que pretende aceder a um recurso;

- recurso a que o utilizador pretende aceder.

Por sua vez, conforme pode ser constatado na figura 3.1, o KDC é composto por dois serviços, o Authentication Service (AS) e o Ticket Granting Service (TGS). O primeiro é

responsável por lidar com a autenticação propriamente dita, enquanto o segundo assume a emissão dos *tickets*, bem como das chaves de sessão.

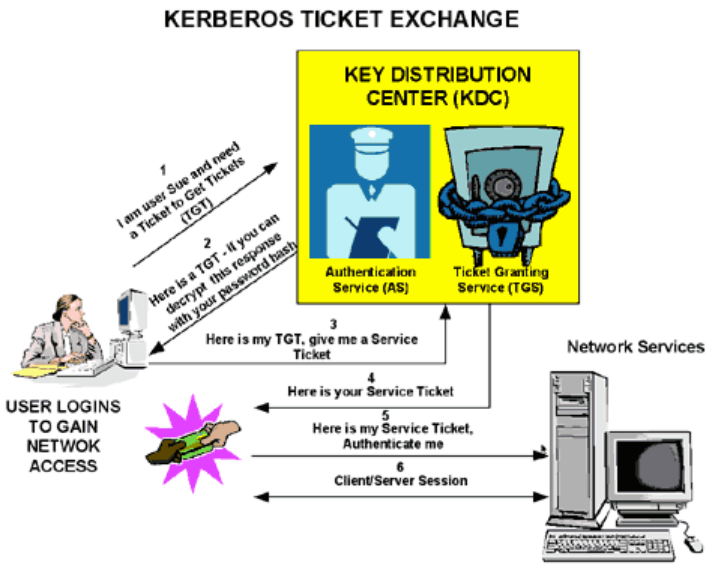


Figura 3.1: Kerberos – componentes e mensagens trocadas entre o cliente, o KDC e o servidor detentor do recurso a que o cliente pretende aceder[9]

Mas antes de vermos mais em detalhe como funciona o Kerberos é importante esclarecermos o conceito de *principal*, e por que é que ele é vulgarmente considerado como sendo a conta de um utilizador. Efectivamente, para um caso particular eles são iguais. Para percebermos em qual, é necessário primeiro que tudo perceber efectivamente o que é o *principal*.

Podemos considerar o *principal* do Kerberos como sendo uma entidade para a qual se consegue emitir *tickets*. É composto por um número variável de componentes que são habitualmente separados por '/'. O formato mais vulgar é *primary/instance@REALM*, em que:

- O *primary* é o primeiro componente. No caso de um utilizador é igual à conta do utilizador. Já no caso de um servidor, toma o valor fixo "HOST".

- O *instance* é opcional e funciona como qualificador do *primary*. É separado deste através do separador "/". No caso de um utilizador é tipicamente nulo. Já no caso de um servidor o seu valor é igual ao nome DNS dele, por exemplo arca.ua.pt.

- O último componente é o REALM e é separado habitualmente do resto do *principal* pelo separador '@'. Frequentemente o REALM é igual ao DNS do nome do domínio, mas em maiúsculas (ex: UA.PT). Este componente pode ser omitido quando o contexto do *principal* for o do KDC.

Agora é facilmente perceptível que no caso de um utilizador cuja conta seja "costa@ua.pt", o seu *principal* poderá ser composto apenas pelo *primary*, e logo ter o mesmo valor desta, ou seja, precisamente "costa@ua.pt".

3.6.1 Kerberos – por dentro do protocolo

O que é guardado na base de dados do KDC não é a *password* do utilizador propriamente dita mas uma chave que é baseada na *password* e no *principal* do utilizador,

$$Kcosta = string2key(Pcosta + \text{“}costa@UA.PT\text{”}),$$

onde Kcosta é a chave, Pcosta é a *password*, “costa@UA.PT” é o *principal* e string2key é uma *hash function*.

Authentication Server Request (AS_REQ)

O cliente envia um AS_REQ ao servidor. Todo o seu conteúdo é enviado sem ser cifrado:

$$AS\_REQ = (PrincipalClient, PrincipalService, IP\_list, Lifetime),$$

onde:

PrincipalClient é o *principal* do utilizador que pretende ser autenticado (ex: “costa@UA.PT”);

PrincipalService é o *principal* do serviço para o qual o *ticket* está a ser solicitado; neste caso é “krbtgt/REALM@REALM”[2];

IP_list é a lista de IPs onde o *ticket* emitido será usado [3];

Lifetime é o período máximo solicitado para a validade do *ticket* a emitir.

Authentication Server Reply (AS_REP)

Primeiro que tudo a base de dados do KDC é usada para confirmar a existência dos dois *principals* indicados no AS_REQ. Se um deles não existir é devolvida ao cliente uma mensagem de erro, caso contrário:

- é gerada uma *session key* (SKTGS) que será usada como segredo partilhado entre o cliente e o TGS;
- é criado o Ticket Granting Ticket (TGT) que é constituído pelos *principals* do utilizador e do serviço (tipicamente krbtgt/REALM@REALM), a lista de IPs - estes três blocos de informação são copiados, sem qualquer tipo de manipulação, do AS_REQ, o *timestamp* (data e hora actual do KDC), o *lifetime* e por fim a *session key* (SKTGS):

$$\begin{aligned} TGT \;=\; & (PrincipalClient, krbtgt/REALM@REALM, IP\_list, Timestamp, \\ & Lifetime, SKTGS) \end{aligned}$$

[2] aparentemente seria supérfluo a indicação do Principalservice no AS_REQ, dado que este seria sempre o mesmo, o principal do TGS: krbtgt/REALM@REALM. Contudo tal não é verdade, dado que é possível a um utilizador não usufruir do SSO e solicitar um único *ticket* para o único serviço a aceder. É assim possível eliminar a solicitação posterior do TGS_REQ, após a recepção do AS_REP;

[3] IP_list pode ser uma lista vazia. Nesse caso o *ticket* poderá ser usado por uma máquina qualquer. Por outro lado, pode conter a lista de IPs de todas as placas de rede de uma máquina. Não é fácil de determinar antecipadamente através de qual dos IP de uma máquina é que será contactado um serviço nela a correr.

- é gerada e enviada a resposta (AS_REP) contendo: o TGT, cifrado usando a *secret key* do serviço (KTGS); o *principal* do serviço, *timestamp*, *lifetime* e *session key*, todos cifrados usando a *secret key* do utilizador (KUser):

$$AS\_REP = \{PrincipalService, Timestamp, Lifetime, SKTGS\}_{KUser}\{TGT\}_{KTGS},$$

onde $\{\ldots\}_K$ significa que a informação entre chavetas está cifrada com a chave K.

Aparentemente a informação existente é redundante. Mas para permitir que cada uma das partes seja decifrada por entidades distintas, a informação tem de ser repetida.

O cliente (de Kerberos) ao receber o pacote AS_REP vai tentar decifrar a parte do ticket que foi cifrada com a chave do utilizador existente no KDC; para isso vai usar a chave que foi obtida usando o *principal* e a *password* que o utilizador forneceu momentos antes. Se o utilizador for quem reclama que é, ou seja, se forneceu uma *password* correcta, a decifra será bem sucedida e por sua vez a *session key* (SKTGS) será extraída e colocada juntamente com o TGT (que permanece cifrado) no repositório de credenciais do utilizador.

Ticket Granting Server Request (TGS_REQ)

Neste momento o utilizador já deverá ter provado quem é (pelo que já possui no seu repositório de credenciais uma SKTGS e um TGT) e pretende aceder a um serviço, mas ainda não possuí um *ticket* para esse efeito. Para o conseguir vai solicitar a sua emissão ao TGT através de um TGS_REQ, construído da seguinte forma:

- primeiro é criado um autenticador (*Authenticator*) com o *principal* do utilizador, *timestamp* da máquina cliente que é cifrado com SKTGS:

$$Authenticator = \{PrincipalClient, Timestamp\}_{SKTGS}$$

- e por fim é criado o TGS_REQ contendo: o *principal* do serviço a aceder e *lifetime*, não cifrados; o autenticador acabado de criar; e o TGT que já está cifrado com a *key* do TGS;

$$TGS\_REQ = (PrincipalService, Lifetime, Authenticator)\{TGT\}_{KTGS}.$$

Ticket Granting Server Replay (TGS_REP)

Quando o TGS_REQ chega ao KDC, mais propriamente ao TGS, este vai verificar que o *principal* de serviço nele indicado existe na base de dados. Se existir, decifra o TGT usando a chave do krbtgt/REAM@REALM e extrai a SKTGS, que por sua vez é usada para decifrar o autenticador. Antes ainda de emitir o *ticket* de serviço as seguintes condições têm de se verificar:

- o TGT não expirou;
- o PrincipalClient constante no autenticador é o mesmo que o existente no TGT;
- o autenticador ainda não está presente na *replay cache* do KDC, nem expirou;

- se a IP_list não é nula é verificado que o IP origem do TGS_REQ é um dos existentes na lista.

Se todas as condições se verificarem isto garante que o TGT realmente pertence ao utilizador que efectuou o pedido, pelo que o TGS pode processar a resposta da seguinte forma:

1. é criada uma *session key* que será o segredo partilhado entre o cliente e o serviço - SKService;

2. é criado um *service ticket*, constituído pelo *principal* do utilizador, o *service principal*, a lista de IPs, o *timestamp* (do KDC), o *lifetime* (o mínimo entre o *lifetime* do TGT e o do *service principal*) e ainda pela *session key* SKService - TService:

$$TService = (PrincipalClient, PrincipalService, IP\_list, Timestamp, Lifetime, SKService)$$

3. é enviada a TGS_REP contendo: o TService, usando a *service secret key* (KService); o *service principal*, o *timestamp*, o *lifetime* e a SKService, tudo cifrado usando a *session key* extraída do TGT:

$$TGS\_REP = \{PrincipalService, Timestamp, Lifetime, SKService\}_{KTGS}, \{TService\}_{KService}$$

Quando o cliente recebe o TGS_REP, usando a SKTGS existente no seu repositório de credenciais, pode decifrar uma parte dele e obter assim a SKService e conjuntamente com o TService (cifrado) guardá-los.

Application Request (AP_REQ)

O cliente sendo detentor de credenciais de acesso ao serviço (*ticket* e *key*) pode finalmente solicitar o acesso ao recurso, recorrendo a uma mensagem AP_REQ. É necessário ter em conta, que ao contrário das mensagens que envolvem um KDC, as AP_REQ não são definidas pelo standard, e variam em função da aplicação/recurso. Quem desenvolveu a aplicação/recurso é que definiu o modo como o cliente deverá fazer prova da sua identidade perante o servidor. Consideremos um exemplo:

- o cliente cria um autenticador que contém o *user principal*, o *timestamp* cifrado com a *session key* SKService, que também será do conhecimento do servidor de aplicação/recurso:
$$Authenticator = \{PrincipalClient, Timestamp\}_{SKService}$$

- é criado o AP_REQ, contendo o *service ticket* TService, cifrado usando a *secret key* KService e o autenticado:
$$AP\_REQ = Authenticator\{TService\}_{KService}$$

Quando o servidor de aplicações/recursos recebe este pedido, vai abrir o *ticket* usando a sua *secret key* e assim obter a *session key* SKService que por sua vez será usada para decifrar o autenticador. De forma a verificar a autenticidade da identidade do utilizador e posteriormente permitir o acesso ao serviço, o servidor irá verificar as seguintes condições:

- o *ticket* não expirou;
- o PrincipalClient presente no autenticador corresponde com o existente no ticket;
- o autenticador não está ainda presente na *replay cache*, nem expirou;
- a IP_list (extraída do *ticket*) é comparada com o IP da origem do AP_REQ. Se não for nula, um dos IPs terá de ser igual;

Conforme se pode facilmente concluir este modo de funcionamento é semelhante ao do TGT quando procede à verificação da autenticidade de um utilizador.

3.6.2 *Pre-Authentication*

A implementação deste mecanismo é opcional no que à norma diz respeito. No entanto a Microsoft, na Active Directory, além de a implementar, obriga ao seu uso. Vamos então perceber porquê.

Como visto anteriormente, a emissão de um *ticket*, entregue via AS_REP, só depende da verificação da existência dos dois *principals* indicados no AS_REQ, na base de dados do KDC: o *principal* do utilizador e do serviço, que tipicamente é o krbtgt/REALM@REALM, quando se trata do TGT. Se o utilizador for ilegítimo o TGT não poderá ser usado dado que não tem forma de usar a chave com que foi cifrado, uma vez que a *password* é desconhecida. Por sua vez não pode ser criado um autenticador.

Apesar disto, o utilizador, mesmo que ilegítimo, ficou na sua posse com um TGT e pode levar a cabo um *brute-force attack* de modo a tentar adivinhar a chave. Esta tarefa não é fácil, mesmo com o poder de cálculo actual, mas é possível. Caso o KDC implemente a *pre-authentication* em modo obrigatório, ao *request* inicial do utilizador irá corresponder um erro a indicar esta obrigatoriedade. Este erro, por sua vez, será tratado como um *soft-error* pelo cliente de kerberos, que irá usar a *password* do utilizador entretanto solicitada, para recorrendo à função string2key gerar uma chave com que vai cifrar o *timestamp*, a enviar no pedido com *pre-authentication*.

Ao receber este pedido o KDC poderá decifrar o *timestamp* usando a chave que sabe do utilizador em causa e por sua vez verificar a autenticidade da identidade do utilizador. O restante processo decorrerá como descrito anteriormente.

Para analisar mais em detalhe o protocolo Kerberos poder-se-á recorrer aos seguintes livros e localizações web: Network Security - Private Communication in a public world (2nd edition) [10], Computer Security (2nd edition) [11], MIT Kerberos Site [12], MIT Kerberos Consortium [13], USC/ISI Kerberos Page [14] e ao RFC 1510 do IETF [15].

3.7 *Single-Sign-On (SSO)*

As implementações de *Single Sign On (SSO)* começaram por ser um esforço que as empresas/instituições levaram a cabo de forma a diminuir a quantidade de identificadores e respectivas *passwords* que um utilizador necessitava de saber e decorar - *Reduced Sign On (RSO)*. Desta forma passou a ser mais aceitável aplicar políticas de *password* mais seguras, mas igualmente mais exigentes para os utilizadores, sem que estes se queixassem tanto. É mais fácil saber e utilizar uma identificação com uma *password* de 12 caracteres do que quatro ou cinco identificações distintas com apenas seis caracteres cada. Desta forma o utilizador recorre ao mesmo identificador e *password* para aceder aos mais diversos sistemas informáticos que a empresa possua. Um benefício que igualmente se obtém é a redução das solicitações nos serviços de *HelpDesk* Informático para atribuições de novas *passwords* aos identificadores dos utilizadores, por estes não se recordarem das actuais.

Entretanto o crescente número de recursos à disposição dos utilizadores, bem como o aumento significativo dos utilizadores existentes vieram colocar os administradores de sistemas perante um novo desafio: "Como seria possível aumentar a segurança, por exemplo através do incremento da complexidade e do comprimento mínimo das *passwords* dos sistemas de autenticação, sem levar os utilizadores a sentirem a necessidade de apontar estas últimas em algum lado para não se esquecerem delas?" Para além disso a solução a implementar deveria possuir custos de implementação baixos e se possível tornar o acesso aos recursos mais facilitado para os utilizadores. Para contornar este problema foi desenvolvido o mecanismo de SSO. Tal como o nome deixa adivinhar, entre outras coisas, ele permite que o utilizador apenas tenha de ser submetido a um único processo de autenticação, após o qual, e durante um determinado período de tempo, não precisa de o voltar a fazer para aceder a qualquer um dos recursos que tem à sua disposição na rede. Passando assim a ter de lidar com menos identidades digitais e a menos frequentemente ter de as apresentar, o utilizador aceita com mais facilidade que o comprimento e a complexidade da *password* seja aumentada.

Toda esta preocupação com o utilizador é devida ao facto, de como já foi referido antes, os seus comportamentos o constituírem como o elemento mais fraco da cadeia de segurança.

A implementação do *SSO* veio também facilitar que na mesma aplicação/sistema informático se recorra a distintas formas de autenticação, em função do nível de risco da informação a que o utilizador se encontra a aceder. O utilizador poderá por exemplo recorrer ao vulgar par identificador e *password* para aceder, mas terá de fornecer outro meio de autenticação quando estiver a aceder a informação mais delicada, a ordenar operações mais importantes, etc. Exemplos de tipos autenticações utilizadas nestes casos são os certificados digitais, os *tokens* de segurança e os dados biométricos.

Vários são os protocolos que implementam *SSO*, mas os mais frequentes de encontrar são o LDAP e o Kerberos.

3.8 Gestão de Identidades

Nesta secção vamos identificar e explicar em que consistem alguns dos sistemas ou mecanismos que permitem facilitar a gestão das Identidades dos utilizadores.

3.8.1 OpenID

O OpenID[16] permite que um utilizador de um novo sítio web use uma identidade que já possui de outro sítio para se identificar. Basta para isso que associe essa identidade ao novo sítio, deixando assim de ser necessário proceder ao preenchimento do formulário, e que passe a gerir mais uma identidade juntamente com as credenciais. Durante o processo de associação, o utilizador tem a possibilidade de identificar quais as propriedades relacionadas com a identidade que pretende que sejam partilhadas com o novo sítio web.

Uma das vantagens da implementação do OpenID é que as credenciais do utilizador são apenas apresentadas perante a entidade que as fornece e nunca aos sítios web de terceiros a que os utilizadores pretendam aceder. Outra, é que muitos dos gestores de sítios web deixam de ter necessidade de gerir identidades, bem como credenciais e problemas inerentes: proceder a alterações de *passwords* entre outros. Por outro lado, muitos utilizadores quando se deparam com uma página de registo de um sítio web acabam por abandonar o acesso que pretendiam fazer. Já acontece o contrário quando são confrontados com a possibilidade de utilização de uma identidade que já possuem.

Estima-se que actualmente existam mais de 50 000 sítios web que usem OpenID e mais de mil milhões de utilizadores registados. De entre as múltiplas entidades que emitem (e simultaneamente aceitam) identidades OpenID as mais conhecidas são as seguintes: Google, Facebook, Yahoo!, Microsoft, AOL e MySpace.

3.8.2 Federação de Identidades

Um mecanismo que veio facilitar a vida ao utilizadores, e por sua vez a dos administradores de sistemas e dos serviços de *HelpDesk*, foi a Federação de Identidades. Caso uma instituição constitua uma federação juntamente com outra, poderá ser permitido aos utilizadores de cada uma o acesso aos recursos da outra empresa, utilizando as credenciais da instituição à qual pertencem. Assim, ao contrário do habitual, em que seria necessário proceder ao registo de cada utilizador na outra instituição, atribuição de credenciais ao utilizador e este tinha de passar a memorizar mais um par de *login* e *password*, basta que um administrador de sistemas dê autorização para que um determinado grupo de utilizadores da outra empresa possa aceder a determinado recurso.

Podem ser constituídos vários tipos de federação, no entanto com o proliferar de sítios e de aplicações web o mais vulgar é serem criadas a este mesmo nível.

Um aliado perfeito das federações de identidades é o *Single Sign On*. Conjuntamente, para além de um utilizador não precisar de recorrer a outra credencial para aceder a recursos de outra instituição, também não precisará de apresentar as suas credenciais em cada vez que tenta aceder a um recurso, mas sim apenas a primeira vez que o faz durante um determinado período de tempo.

De entre as várias implementações do conceito de Federação de Identidades que surgiram até hoje podemos destacaram duas: *Active Directory Federation Services (ADFS)* e *Shibboleth*. A primeira mais não é do que a proposta da Microsoft, que consiste na federação de dois ou mais domínios Active Directory, através de servidores de ADFS, que servem de emissores e de verificadores de *claims* emitidas para permitir (ou não) o acesso a servidores web (também Microsoft). Já a segunda, é um projecto da Internet2 que tentou com base em SAML criar uma implementação *open-source* de um mecanismo web de autenticação, com SSO. Dadas as especificidades bem como o actual estado de permanente evolução do Shibboleth, faz todo o

sentido abordá-lo mais aprofundadamente na sub-secção seguinte.

3.8.3 Shibboleth

O projecto Shibboleth[17] iniciou-se em 2000, com o propósito de resolver os problemas de partilha de recursos entre organizações cujas infraestruturas de autenticação e autorização pudessem ser completamente diferentes. Após várias versões beta em Julho de 2003 surgiu o Shibboleth 1.0; em Agosto de 2005 a versão 1.3; e em Março de 2008 a versão 2.0.

Como o Shibboleth é uma implementação web de acSAML, ele mantém o conceito de *Identity Provider (IdP)* e de *Service Provider (SP)*. Esta duas componentes são respectivamente o emissor e verificador de identidades, e o servidor web. O seu funcionamento baseia-se nas seguintes premissas:

- O utilizador só é registado uma vez, pela organização/instituição à qual o utilizador pertence (*Home organization*). Esta é responsável por manter actualizada a informação dos utilizadores bem como gerir as credenciais. Exemplo de instituições deste tipo são Universidades, ISPs, etc.
- A autenticação é sempre efectuada pela e na Home organization (via IdP). Para além da simples autenticação, poderá ser fornecida informação adicional sobre o utilizador em causa, quando tal seja solicitado pelo recurso ao qual o utilizador está a tentar aceder e caso este autorize.
- A autorização de acesso ao recurso é da responsabilidade exclusiva do recurso (SP), baseado-se na informação obtida sobre o utilizador. Deste modo a gestão federada de identidades assenta no conceito que os recursos confiam no sistema de autenticação da *Home organization* do utilizador e que vão usar informação que este lhes vai fornecer para decidirem sobre a componente de autorização.

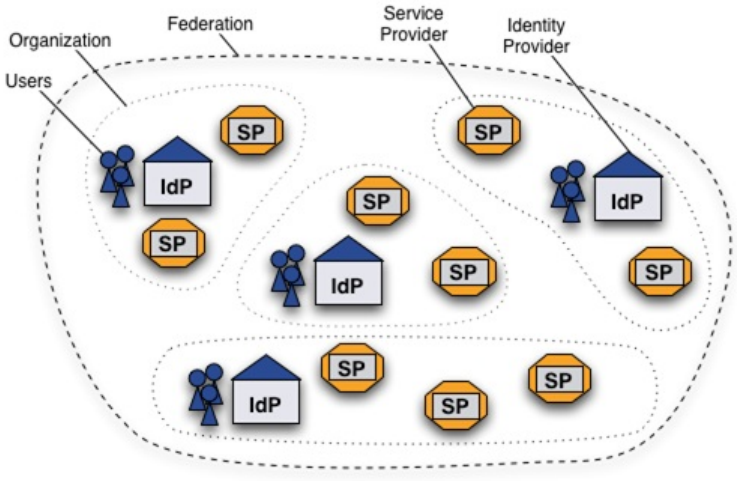


Figura 3.2: Exemplo de uma Federação constituída por quatro instituições, detentoras de *Identity Providers* e *Service Providers*[17]

De forma sucinta, de seguida tenta-se explicar como funciona o processo de acesso a um *Service Provider*.

Processo de acesso a um *Service Provider*

Para melhor se perceber o que acontece em cada fase, vamos concretizar para o acesso ao sítio moodle.ua.pt, ou melhor, https://moodle.ua.pt/secure, uma sub-área dele que está protegida por uma sessão Shibboleth válida.

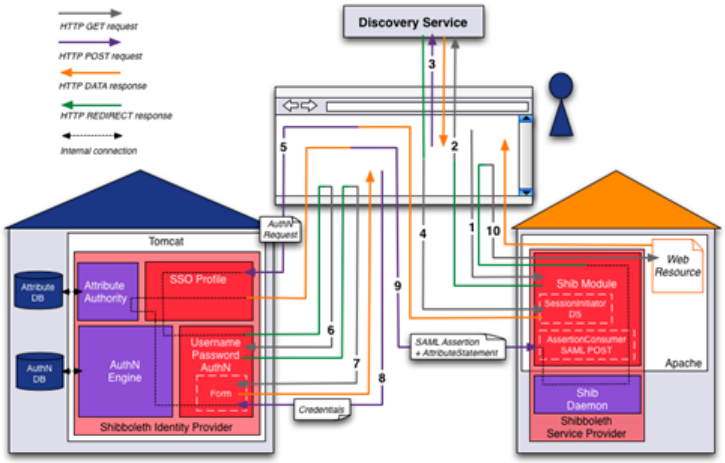


Figura 3.3: Shibboleth - Funcionamento[17]

1. O utilizador usa um cliente web (*browser*) e acede a um SP, localizado em `https://moodle.ua.pt/secure`

   Dado que `https://moodle.ua.pt/secure` está protegido pelo Shibboleth SP, vai ser testado se o utilizador já possui uma sessão Shibboleth, ou seja, se já se autenticou anteriormente. Se o utilizador ainda não está autenticado, o servidor web vai responder com um *HTTP Redirect* para o *Discovery Service (DS)* localizado em `http://idp.ua.pt/discovery`. Uma vez que o serviço DS precisa de saber posteriormente para onde deverá redireccionar o *browser* depois do utilizador escolher a sua *Home Organization*, a informação é fornecida através de *GET parameter*

2. O *browser* consequentemente envia um pedido ao DS

   O DS responde com a página que permite ao utilizador escolher o IdP, tal como na figura 3.4.

3. Na página do DS, o utilizador submete a selecção do IdP

   O DS devolve um *redirect* cujo destino foi indicado como *return*, incluindo também a informação do IdP seleccionado.

4. Devido à resposta anterior do *redirect*, o *browser* do utilizador vai enviar um pedido de criação de sessão.

   O *session initiator* recebe-o e cria um *authentication request* que vai devolver dentro de um *auto-submit-post-form*

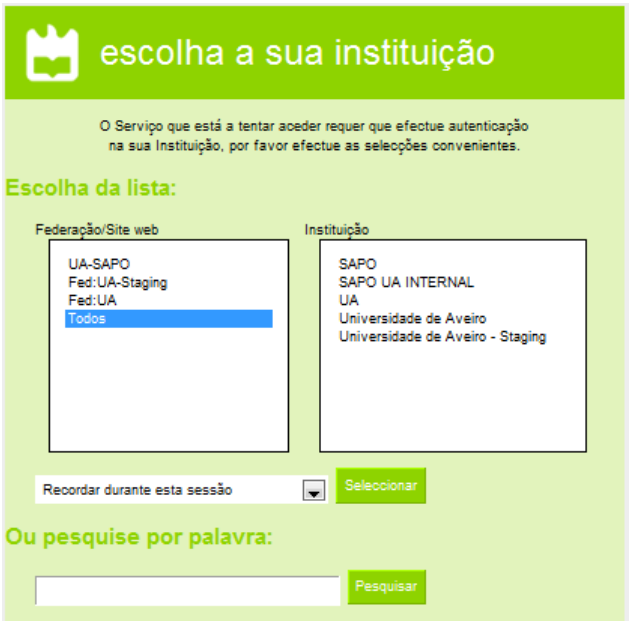


Figura 3.4: *Discovery Service* da UA[18]

---

5. O *browser* entrega um *request* SAML, com recurso a Javascript

   O IdP verifica o *authentication request*, dado que o utilizador não está autenticado ele vai enviar um *redirect* para o *login handler* apropriado (neste caso: *username/password*)

6. O *browser* vai ser redireccionado para o *login handler username/password*:

   O IdP envia então um *redirect* para a página específica de *username/password*.

7. Em seguida o *browser* envia um *GET request* para a página de *username/password*

   O servidor web responde com a página de *username/password*, tal como constante na figura 3.5

Figura 3.5: *Identity Provider* da UA[19]

8. O utilizador escreve o seu *username* e *password* e submete-as ao IdP

O mecanismo de autenticação do IdP verifica as credenciais. Depois do utilizador ter sido autenticado, a *attribute authority* proceder ao *attribute resolving* e ao *attribute filtering*. Em seguida é gerada uma página HTML que inclui uma SAML *assertion*. Dado que esta *assertion* contêm não só um *authentication statement* mas também um *attribute statement* com os atributos do utilizador, é designada "Attribute Push". A *assertion* é enviada recorrendo a um *auto-submit-post-form*

9. De seguida o *browser* envia o seguinte *request*

   O SP processa a SAML *assertion* que inclui o *authentication* e *attribute statements*. Por fim envia um *redirect* para o recurso que o utilizador pretendia aceder, cujo URL foi armazenado no *cookie-shibstate*

10. Tal como no passo 1, o *browser* faz o *request* do recurso protegido pelo Shibboleth

    `https://moodle.ua.pt/secure`

    Desta vez contudo, o utilizador está autenticado. Para decidir quanto à autorização de acesso ao recurso, o módulo *mod-shib* que está embebido no servidor web examina as Shibboleth *access rules* e tenta cumpri-las recorrendo aos atributos do utilizador. Adicionalmente usa informação sobre a autenticação propriamente dita: momento, método usado, etc. Após ser autorizado acesso o conteúdo da página é devolvido ao *browser*

3.8.4 Cartão de Cidadão

O Cartão de Cidadão é um cartão de identificação dos cidadãos da República Portuguesa, pois possui informação impressa nas faces que permitem ser usado como tal, e simultaneamente um *smart-card*. Possui impressas e guardadas electronicamente informações de carácter público (números de identificação do próprio cartão, de contribuinte, de utente dos serviços de saúde, da segurança social, de eleitor, etc), mas para além destas existem outras, armazenadas no último formato, cujo acesso está protegido por PIN, como por exemplo os dos dois certificados digitais nele contidos (um para assinatura, o outro para autenticação). Com o certificado digital de autenticação é possível fazer prova da identidade electrónica do cidadão ao qual o cartão pertence. Já com o certificado digital de assinatura é possível certificar que determinado documento é da autoria de um determinado cidadão. No Cartão de Cidadão estão ainda presentes minúcias de duas impressões digitais, que permitem autenticar ou não o portador do cartão. De realçar que a verificação é efectuada tendo por base a informação obtida por um leitor de impressões digitais, que é posteriormente confrontada por uma aplicação residente no chip do cartão com a informação nele mesmo contida. Desta forma a informação contida no cartão e que diz respeito às impressões digitais nunca flui para o exterior do mesmo.

Capítulo 4

Autenticação na Universidade de Aveiro

Neste capítulo vamos começar por caracterizar a autenticação na Universidade de Aveiro (UA), identificando os mecanismos de autenticação existentes, as aplicações e os sistemas em que eles são usados e ainda qual o âmbito de utilização. Em seguida, vão ser referidas propostas de melhoria que se justificam na sequência de um estudo mais profundo das tecnologias envolvidas nos sistemas, aplicações, ou serviços existentes.

4.1 Sistema central de autenticação

O Centro de Informática e Comunicações da Universidade de Aveiro (CICUA) até meados no ano 2000 administrava em cada departamento uma solução de partilha de ficheiros, partilha de impressoras, bem como de correio electrónico. A autenticação e autorização destes serviços baseavam-se em sistemas Novell ou Windows NT 4 e as soluções de correio electrónico tipicamente eram suportadas por Mercury em Novell (com cliente Pegasus Mail) ou Microsoft Exchange.

Pela dimensão, dificuldade de administração, limitações destas soluções, entre outras, o CICUA optou por implementar uma plataforma de correio electrónico central, que fosse transversal a todas as unidades da UA. Essa plataforma baseava-se numa solução da empresa Stalker designada de CommuniGate Pro e, à parte actualizações entretanto efectuadas, ainda é a solução hoje existente.

Os sistemas de autenticação existentes na altura tinham várias limitações, nomeadamente por serem soluções isoladas, ou seja, sem qualquer tipo de relação entre elas, o que implicava que todas as pessoas que usavam computadores em duas ou mais Unidades, tinham de possuir um *login/password* em cada sistema de cada Unidade. Dado isto, o CICUA decidiu implementar uma plataforma *Active Directory*, com um domínio de topo ua.pt e vários sub-domínios, um por cada Unidade (ex: cic.ua.pt). No caso das Unidades que possuíam servidores Windows NT 4, o repositório de utilizadores foi migrado, permitindo assim que os utilizadores preservassem as credenciais que já detinham. Nas restantes foram entregues credenciais novas a cada pessoa. Entre outras coisas, quando era configurado para tal (dada autorização), passou a ser possível a utilização de uma credencial de uma unidade para aceder a recursos de outra (ex: ccosta@cic.ua.pt para aceder a um computador do Departamento de Ambiente e Ordenamento).

No ano lectivo de 2002/2003, optou-se por gerir de forma integrada os laboratórios de informática de toda a UA. Para isso, foram criados o domínio alunos.ua.pt e as contas de utilizadores para todos os alunos e docentes no domínio ua.pt (ex: a12457@ua.pt e f1976@ua.pt). Com elas passou a ser possível, quer a alunos quer a docentes, acederem aos computadores dos vários laboratórios existentes nas várias Unidades, recorrendo à mesma conta de utilizador. Esta veio a comprovar-se como uma solução eficaz para a gestão integrada dos recursos das várias unidades, sendo conseguida a redundância com dois servidores que asseguravam o domínio, algo que antes só seriam possível com a duplicação de todos os existentes.

A pouco e pouco foram surgindo serviços geridos pelo CICUA que foram disponibilizados às pessoas que trabalham ou estudam na UA cuja autenticação reside no domínio ua.pt. Foi assim eliminada a existência de credenciais individuais para cada serviço novo. Para além do caso já exposto dos laboratórios de informática, foram disponibilizados serviços como os acessos remotos (vpn.ua.pt), a rede sem fios (eduroam), entre outros.

Até 2007, os utilizadores tinham de se registar (e fornecer os seus dados de identificação) junto de cada Unidade/Serviço que disponibilizava sistemas de informação. Caso este tivesse vários tipos de vínculo, teria de fornecer várias vezes a mesma informação, pois não havia qualquer tipo de intercâmbio de uns sistemas para os outros. Consequentemente, para cada sistema informático ao qual fosse disponibilizado acesso personalizado tinham de ser criadas e geridas credenciais, o que implicava por exemplo que no acesso ao Portal Académico (vulgo PACO, ou Secretaria Virtual) tivesse um par de login e password, no caso do e-mail tivesse outro, para acesso aos computadores dos laboratórios outro (AD @ua.pt), para acesso aos computadores departamentais outro, para requisição de livros na Biblioteca outro, etc. Para tentar eliminar todas estas credenciais, simplificar a convivência dos utilizadores com os sistemas informáticos da UA, e agilizar o funcionamento dos sistemas de bases de dados, foi concebido o sistema Registo Central de Utilizadores (RCU). Os objectivos principais deste eram:

- criar uma identidade electrónica única para cada pessoa/utilizador.
- recurso às mesmas credenciais para acesso aos sistemas informáticos de uso generalizado dos utilizadores (cujo login foi designado por Utilizador Universal - UU)
- personalização da conta de utilizador com base no seu nome (ex: costa@ua.pt)
- indexação dos dados das diferentes bases de dados a uma chave imutável
- partilha de dados pessoais cujo interesse seja comum a vários serviços (nome, morada, documentos de identificação, etc)
- notificação de determinados serviços/aplicações informáticos/as do registo de um novo utilizador ou da alteração do registo já existente
- permitir a autenticação com o UU em sistemas que não consigam interagir com a AD (LDAP ou kerberos), recorrendo por exemplo a *webservices*

Em Junho de 2007, foi encetado um processo designado por Validação de Identificação Pessoal (VIP). Nele as pessoas com vínculos activos com a UA à data eram convidadas a verificarem, corrigirem e validarem as suas credenciais de sistemas informáticos e os seus dados pessoais que anteriormente foram coleccionados junto das várias fontes de dados[1] que

[1]exemplo de fontes de dados: Direcção dos Recursos Humanos (DRH), Serviços Académicos (SACAD)

vinculam pessoas à UA. Após a entrada em funcionamento do RCU as várias credenciais seriam abandonadas em detrimento do recurso ao UU. Durante o VIP, os utilizadores da UA eram também convidados a indicar uma lista de UUs pretendidos. No fim do processo foram atribuídos os UUs com base nas prioridades estabelecidas e nas preferências indicadas, e com base neles renomeadas as contas e áreas existentes, e ainda reconfigurados os serviços que possuíam credenciais próprias para passarem a usar os UUs.

Desde então, os utilizadores com os mais diversos tipos de vínculos, sejam eles de funcionários (docentes, não docentes), prestadores de serviços, bolseiros da UA, bolseiros externos, ou outros, passaram a deter um único UU, mesmo que possuam simultaneamente vários vínculos. Por exemplo, desde Setembro de 2007 um caloiro que entra na UA, quando procede à sua matrícula, para além de fornecer os seus dados e documentos, tem uma tarefa adicional: escolher o seu Utilizador Universal. Mas com este simples passo dado por ele, conjuntamente com o registo dos dados do utilizador junto do RCU para além dos sistemas dos próprios Serviços Académicos, veio permitir que o utilizador deixasse de ter de se dirigir aos vários serviços da UA a solicitar o acesso e credenciais para cada sistema, como acontecia até então, e pudesse no dia seguinte, com recurso a um único identificador (o seu UU), aceder a todos os serviços que foram considerados de uso generalizado:

- plataforma de e-mail
- rede sem fios - eduroam
- Arquivo Central de ficheiros - ARCA
- Portal Académico Online - PACO
- plataforma de e-Learning
- requisição de livros na Biblioteca - aleph
- laboratórios de informática
- portal integrado my.ua.pt
- Tele-Trabalho - vpn.ua.pt

Caso este aluno já possuísse um vínculo activo, imaginemos de funcionário não docente, então não seria solicitada a escolha de um UU, mas sim o reconhecimento de que o utilizador em causa já possuía UU. O novo vínculo de aluno iria atribuir ao UU em questão um conjunto novo de serviços que o utilizador teria para além dos respeitantes ao vínculo de funcionário.

Já no ano de 2008 foi iniciado um longo processo de migração dos computadores dos domínios departamentais para o domínio clients.ua.pt, acompanhado da migração dos utilizadores usados para acesso aos mesmos, passando a usar-se também o UU. Neste processo tem vindo também a eliminar-se as soluções de partilha de impressoras e de ficheiros que eram assegurados pelos controladores de domínio dos domínios departamentais, para se adoptarem os sistemas centrais printers.servers.ua.pt e arca.ua.pt, respectivamente.

serviço/sítio web	conta	autenticação	Observações
Mail departamental	local	proprietária do CommuniGate	-
PACO	local	BD	visitantes
aplicação tipo 1	mista	AD, via ws do RCU/Oracle	-
aplicação tipo 2	local	Oracle	-
sítio web tipo 1	local	BD/ficheiro	HTTP/HTTPs
sítio web tipo 2	UU	AD	HTTP/HTTPs
sítio web tipo 3	UU	AD, via ws do RCU	HTTP/HTTPs
cliente Linux 1	UU	AD	autorização via LDAP não AD
cliente Linux 2	servidor	NIS	-
cliente Linux 3	local	passwd	-

Tabela 4.1: Sistemas de autenticação e tipos de contas

4.2 Outros sistemas de autenticação

Na Universidade de Aveiro há uma panóplia de serviços, aplicações e sistemas informáticos muito mais vasta do que os descritos anteriormente de utilização generalista. É o caso de aplicações ou sítios web utilizados por grupos de pessoas restritos, nomeadamente, pelos funcionários dos Serviços Académicos e das secretarias departamentais, dos Serviços Financeiros e de Património, do CICUA, dos Serviços de Documentação, entre outros, mas também sítios web de âmbito departamental ou de grupos de trabalho, ou outros cuja utilização é mais pontual.

Na tabela 4.1 estão identificados alguns dos sistemas de autenticação usados e caracterizados no que diz respeito ao tipo de contas e de sistemas de autenticação, entre outros pormenores. Conforme pode ser observado existem vários outros mecanismos de autenticação em uso na UA para além do anteriormente designado por sistema central. Desde contas cujas credenciais residem em ficheiros, passando por alternativamente se encontrarem guardadas em bases de dados, noutros casos recorrendo a mecanismos proprietários, como é o caso do Oracle, ou do uso de *webservices* desenvolvidos no âmbito do projecto RCU, entre outros. Com estes últimos foi possível permitir que algumas aplicações sem qualquer tipo de suporte nativo para usarem autenticação LDAP ou kerberos o passassem a conseguir fazer através de uma aplicação com a qual falam através de *webservices*, e que vai junto da AD, em nome da primeira efectuar a autenticação. Temos ainda os clientes linux, que se podem subdividir essencialmente em três tipos:

1. laboratórios de informática da UA – apesar da autenticação recorrer a kerberos (AD), a autorização não é efectuada com base na AD, mas sim num outro servidor ldap (linux), por inexistência de alguns dos atributos das contas dos utilizadores que são essenciais.

2. laboratórios do DETI (alguns) – por necessidades específicas inerentes às disciplinas leccionadas em alguns laboratórios do DETI, os computadores neles existentes recorrem a um servidor NIS para efectuarem a autenticação, bem como a autorização.

3. clientes isolados/pessoais – neste cenário a autenticação baseia-se nos ficheiros locais /etc/users e /etc/passwd, bem como a autorização de acesso aos recursos localmente existentes no ficheiro /etc/groups.

Característica	Nova política
histórico de password	5
idade máxima da password	180
idade mínima da password	0
comprimento mínimo da password	8
password com complexidade	sim

Tabela 4.2: Nova política de passwords

Para além dos diversos mecanismos de autenticação usados, cujas características essenciais foram abordadas na secção 3.5, há igualmente que referir que existem protocolos de comunicação com níveis de segurança bastante distintos. Por exemplo, um utilizador quando envia as suas credenciais para o sítio web a que se encontra a aceder e este não recorre a HTTPS, está a enviá-las em texto simples. Muitas vezes ele não tem noção disso, e que assim é possível e fácil que as mesmas possam ser capturadas por terceiros e posteriormente usadas no acesso ao mesmo sítio web, ou até a outros que apesar de já utilizarem HTTPS, possuem as mesmas credenciais do sítio web cujo protocolo é vulnerável.

Se extrapolarmos os casos tipo que são apresentados na tabela 4.1 para a dimensão da UA podemos concluir facilmente que é bastante elevado o número de sítios web e de aplicações com autenticações locais ou que apesar de recorrerem à autenticação central o fazem com recurso a protocolos de comunicação intermédios vulneráveis.

Em seguida inicia-se a abordagem às propostas que eventualmente se mostram necessárias, seja para melhorar a segurança, ou a usabilidade por parte dos utilizadores, de serviços/sistemas centrais, departamentais, ou mesmo individuais, quer no caso da realidade do Campus de Santiago, quer nos pólos da UA dispersos pelo Distrito de Aveiro.

4.3 Política de passwords

Tal como foi referido na secção 3.3 o tipo de autenticação mais vulgar é o que se baseia em *login/password*, e o caso da UA não é excepção. No que diz respeito ao sistema central de autenticação a política de *passwords* é algo conservadora: complexidade e número de caracteres baixo, acrescido do histórico apenas da *password* anterior. Identificada como melhorável, está em vias de implementação uma nova política a aplicar às contas dos utilizadores em geral. As principais características estão resumidas na tabela 4.2

Associadas às políticas de passwords propriamente ditas, costumam surgir as políticas de bloqueio de contas que tentam evitar que as contas sejam alvo de quebras por terceiros. Isto é vulgarmente conseguido definindo que uma conta será bloqueada no caso de ocorrer um certo número de falhas de autenticação por password errada, num determinado período de tempo, e ainda durante quanto tempo estará a conta bloqueada.

A activação da nova política de password deverá ser efectuada de modo faseado, de modo a permitir encontrar eventuais limitações de aplicações, páginas web, ou mesmo de sistemas/servidores e resolvê-las com um impacto mínimo para os utilizadores.

Posteriormente, e dado que a actual versão da Active Directory (2008) o permite, ao contrário das anteriores, poderão ser implementadas políticas de password distintas para

outros tipos de contas, nomeadamente contas de administração e de serviços (por exemplo backups).

4.4 Kerberos

O Kerberos foi identificado na secção 3.5 como sendo o protocolo de autenticação mais completo e seguro, pois oferece autenticação mútua e cifragem, permite o recurso a autenticação forte e a delegação, e possui ainda a vantagem da password nunca fluir na rede. A juntar a isto, é uma norma e é independente de plataformas, fazendo jus ao lema do Kerberos consortium[13]: "Our mission is to establish Kerberos as the universal authentication platform for the world's computer networks". A implementação Microsoft permite ainda a implementação da autorização de acesso a recursos e a coexistência de um KDC com um servidor de LDAP baseados na informação do directório.

Desta forma, o Kerberos deverá, sempre que possível ser o protocolo de autenticação de eleição. Existem na UA alguns serviços que são candidatos a recorrerem a este protocolo, nomeadamente: os serviços de partilha de ficheiros (ARCA) e de impressão (Printers). Actualmente os servidores que suportam estes serviços fazem parte do domínio servers.ua.pt, cujos KDCs não são contactáveis nas redes dos computadores clientes da UA, pelo que o protocolo de autenticação e autorização usado é o NTLM. Para passar a ser possível a emissão dos tickets Kerberos de serviço é necessário que os KDCs estejam contactáveis, como é o caso dos que asseguram o domínio clients.ua.pt. Assim basta que os servidores que suportam os serviços identificados sejam migrados para o domínio clients.ua.pt para que passam a usufruir do protocolo Kerberos.

A migração de domínio referida no parágrafo anterior pode ainda assim não ser suficiente no caso da partilha de impressão para alguns tipos de clientes, dos quais destaco os das distribuições Linux e Macintosh. Para estes uma alternativa poderá estar na disponibilização do serviço de impressão via *Internet Printing Protocol* (IPP).

4.5 *Lightweight Directory Access Protocol – LDAP*

O *Lightweight Directory Access Protocol (LDAP)* é o protocolo por excelência da autorização. Esta é conseguida com base da informação do directório que está na sua retaguarda. No caso do sistema central de autenticação e neste caso de autorização, o repositório é o *Active Directory*. Esta implementação da Microsoft possui a particularidade de ser *multi-master*, ou seja, com excepção de alguns papéis (*roles*) muito específicos que são assegurados apenas por um dos servidores que a constituem tudo o resto é indiferentemente efectuado por qualquer um deles, fazendo assim com que seja considerada um solução de muito alta disponibilidade e resiliência. Como exemplo disto mesmo pode ser dado todo o processo de actualização para 2008 da plataforma que assentava ainda em servidores Windows 2000 e que foi terminado recentemente sem que tivesse ocorrido qualquer tipo de quebra de serviço.

Na tabela 4.1 pode ser constatado que existem clientes linux que usam um outro serviço de LDAP que não o central. Isto devia-se à inexistência no LDAP do sistema central de alguns atributos essenciais para que estes clientes efectuassem a autorização. Esta situação foi revista, efectuando uma alteração na estrutura de dados do *Active Directory* para que estes mesmos atributos passassem a estar disponíveis através do LDAP. Após isto foi possível reconfigurar os clientes em causa, nomeadamente os existentes nos laboratórios de informática

da UA, de forma a recorrem ao sistema central, tal como já faziam no que dizia respeito à autenticação (via kerberos).

4.5.1 LDAP com SSL

Pela facilidade de utilização do LDAP este conseguiu convencer os programadores a usarem-no massivamente e a adoptarem-no como protocolo de eleição, mesmo para o que ele não havia sido pensado: a autenticação. De facto o LDAP possui nativamente um método de autenticação que de tão elementar que é foi logo designado de *simple* (simples). Isto faz com que o trabalho do programador seja facilitado pois só tem de lidar com um protocolo seja para obter dados para a autorização de um utilizador, ou para saber se as credenciais que o utilizador forneceu eram válidas. No entanto, ao ser solicitado ao servidor de LDAP que teste as credenciais estas são-lhe enviadas em texto cru, conforme pode facilmente ser observado fazendo uma captura dos pacotes trocados, tal como mostra a figura 4.1. É assim muito fácil a alguém mal intencionado capturar as credenciais e posteriormente fazer uso delas para o que bem entender.

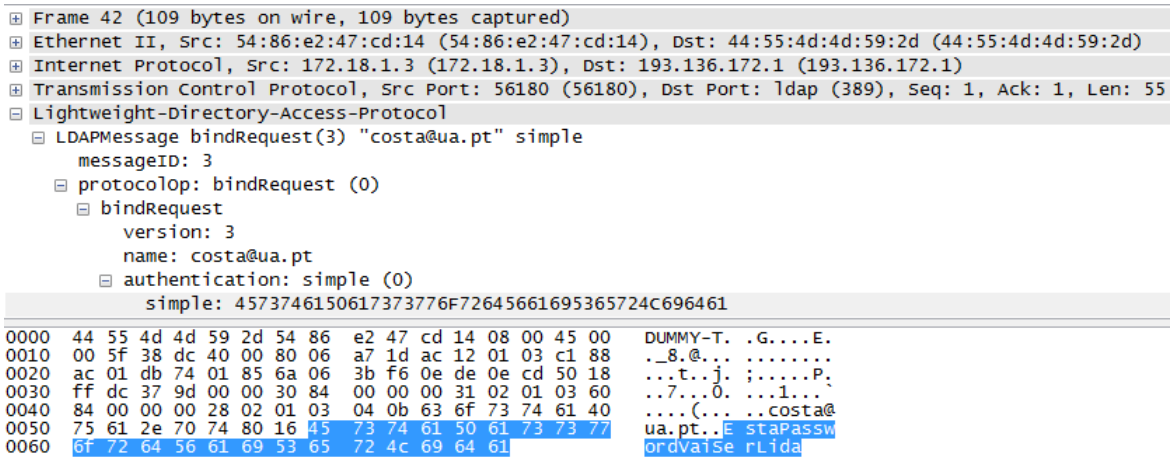


Figura 4.1: Captura de um *bind* LDAP com recurso ao método *simple*

Existem três formas de resolver este problema:

1. banir por completo o uso de LDAP para autenticação de utilizadores, forçando os clientes deste serviço a utilizarem Kerberos;

2. restringir a autenticação de utilizadores via LDAP apenas a métodos mais seguros (que não o *simple*), por exemplo recorrendo ao *Generic Security Service Application Program Interface (GSSAPI)* para autenticar via Kerberos;

3. configurar o LDAP para operar sobre SSL.

Esta última hipótese é talvez a que seja mais facilmente aceite pelos programadores por ser a que é mais transparente de adoptar por estes. Pelo que a recomendação vai no sentido de dotar o Active Directory de SSL.

Um servidor de LDAP pode em opção efectuar *signing* (assinatura) dos dados enviados para o cliente, o que permite a este último não a garantia de que a informação recebida não seja adulterada nem perceptível por outrem, mas sim que a sua integridade não seja

comprometida sem que tal não seja detectado. Neste caso o cliente pode decidir se acredita ou não nos dados recebidos para efectuar a autorização, mas não tem a garantia de que as credenciais usadas na autenticação não possam ser comprometidas. Esta potencialidade tipicamente só é usada quando os clientes são exclusivamente Microsoft, pois nem todos os restantes a suportam (ex: Macintosh e Linux).

4.6 Sítios web

O número e a diversidade de tipos de sítios web existentes na UA é bastante significativo. A nível de segurança as necessidades de uns com certeza serão diferentes das de outros. Mas podemos tentar agrupá-los no que diz respeito ao uso de autenticação e à criticidade/integridade dos conteúdos:

1. sem autenticação e sem conteúdos críticos
2. com autenticação e sem conteúdos críticos
3. sem autenticação e com conteúdos críticos
4. com autenticação e com conteúdos críticos

O primeiro grupo não requer qualquer tipo de cuidado especial.

Já o segundo apesar de os conteúdos que serão acedidos não serem críticos requerem autenticação. Habitualmente quem desenvolve o sítio web tem de gerir na totalidade as credenciais dos utilizadores; em alternativa pode aceitar credenciais de algum sistema de autenticação e ir junto dele verificar se as mesmas são válidas (usando para isso Kerberos ou LDAP). No caso deste último sistema ser o central da UA, para não haver a possibilidade de as credenciais serem comprometidas convém que a comunicação cliente servidor web seja cifrada, ou seja que recorra a SSL, e por sua vez que use o protocolo HTTPS. De qualquer dos modos o programador tem na mesma de tratar da autorização, assim como de obter as informações relativas ao utilizador para decidir sobre a mesma. Todo este trabalho do programador pode ser simplificado se recorrer ao Shibboleth, ainda com a vantagem do utilizador poder usufruir de *Single Sign-On* e de não ter de confiar em mais um sítio web para entregar as suas credenciais e acreditar que as mesmas não serão alvo de uso malicioso. Deixa assim também de haver a necessidade de gerir o certificado que seria usado pelo protocolo HTTPS.

No caso do terceiro grupo, uma vez que os conteúdos são críticos, os sítios web deverão usar o protocolo HTTPS.

Relativamente ao quarto e último grupo, as necessidades acabam por ser uma conjugação das existentes no segundo e terceiro grupo, pelo que o recomendável é que se opte por usar o protocolo HTTPS para a componente de comunicação e a Shibboleth para a autenticação e a obtenção de dados sobre o utilizador para serem tomadas as decisões de autorização.

Deverá assim ser encetado um esforço conjunto dos gestores dos servidores onde os sítios web se encontram alojados e dos programadores dos sítios para alterarem estes últimos de modo a que recorram ao Shibboleth, podendo assim os utilizadores usufruir das vantagens deste.

O CICUA como entidade responsável pelos principais servidores web e pelo IdP do Shibboleth deverá disponibilizar aos potenciais interessados neste sistema a documentação necessária para que a sua utilização seja agilizada e facilitada tanto quanto possível. Uma proposta de

guião à instalação da componente de *Service Provider* do Shibboleth (que reside no servidor onde está alojado o sítio web) está presente neste documento em apêndice.

À semelhança do que foi feito com os sítios web que passaram a utilizar o sistema de autenticação central (UU), que têm na sua página de autenticação a indicação de que estão integrados com o Utilizador Universal, talvez fosse de criar um logótipo que seria atribuído aos sites que fossem considerados seguros. Eventualmente com mais do que um nível de segurança, que seria aferido em função de alguns parâmetros a validar, como se as credenciais são fornecidas ao próprio sítio/aplicação web ou se usa alguma mecanismo de federação, se usa SSO, se usa HTTPS, etc. Complementarmente poderia ser criada e disponibilizada numa página web uma lista com as aplicações/sites web que fossem sendo distinguidos com o logótipo, incentivando assim quem efectua desenvolvimento a fazê-lo da melhor forma, pelo menos no que à autenticação diz respeito.

4.7 *Internet Protocol Security – IPSec*

O *Internet Protocol Security (IPSec)* recorre a vários protocolos normalizados de forma a garantir que, ao nível da camada IP, a troca de pacotes seja segura. Isso é conseguido através da autenticação e da cifragem dos pacotes IP que são trocados entre dois computadores (cliente e servidor), dois equipamentos de rede (*router*, *gateway*, ou *firewall*), ou um de cada.

Segundo [20], "(...)if IPSec is implemented in end systems, upper-layer software, including applications, is not affected", ou seja, pode recorrer-se ao IPSec no caso em que não seja possível alterar as aplicações para que estas usem protocolos de autenticação ou de comunicação mais seguros, fazendo-o assim de forma transparente. No entanto tal só é viável quando o número de clientes envolvidos não for elevado e seja fixo, ou seja, a aplicação encontra-se instalada e vai ser usada sempre nos mesmos computadores. Isto porque, num novo computador, para além da instalação da aplicação propriamente dita, é também essencial que seja gerado e posteriormente instalado o certificado do cliente no servidor e o certificado do servidor no cliente.

Em alguns casos, a utilização do IPSec poderá ser substituído pela utilização de ligações VPN. Nestas, o tráfego entre o cliente e o servidor de VPN é igualmente cifrado, restando como vulnerável apenas o troço de rede que estiver entre o servidor de VPN e o computador com o qual se pretende efectivamente comunicar.

4.8 Aplicações executadas remotamente

O IPSec foi apresentado anteriormente como um mecanismo ao qual se poderia recorrer no caso da existência de aplicações cujas componentes de autenticação e/ou comunicação fossem débeis e cuja alteração não fosse viável. Mas existem outras alternativas. Uma delas é a execução remota de aplicações que se explica de seguida.

O conceito é já bastante antigo: terminais que apenas mostram informação no ecrã e procedem à leitura dos sinais vindos do teclado e do rato; o processamento não é local, mas sim central, realizado por um servidor ou grupo de servidores. Antigamente estes terminais eram muito básicos, com capacidades gráficas reduzidas essencialmente a texto, no entanto, com a diminuição significativa dos custos de aquisição de um computador, estes terminais foram sendo abandonados e as aplicações foram sendo desenvolvidas ou remodeladas de modo a explorarem as novas capacidades gráficas e computacionais ao seu dispor.

A execução remota de aplicações que funcionem em modo gráfico não é possível recorrendo a protocolos como o telnet, ou similares. As alternativas estão em protocolos como o *Remote Framebuffer* (RFB) usado no *Virtual Network Computing* (VNC), no *Remote Desktop Protocol* (RDP), ou no X Window Protocol que permitem que sejam visualizadas sessões de terminal com capacidade gráfica e fluidez tais que proporcionam uma experiência de utilização como se a execução da aplicação fosse local. Entre os vários produtos existentes no mercado destacam-se o Presentation Server da Cytrix e o Terminal Server (TS) da Microsoft como sendo os que têm maior implantação. No caso deste último, na sua versão 2008 tem uma funcionalidade que permite o estabelecimento de uma sessão de terminal, não para acesso pleno à sessão como era até então sempre usado, mas sim para a execução de apenas uma aplicação. O resultado final é idêntico ao obtido na execução de aplicações que se encontram efectivamente instaladas no computador, e tal como estas, está à distância de um duplo-clique sobre um atalho. Esta funcionalidade é designada de RemoteApp.

À RemoteApp pode ser associada a uma outra potencialidade: Terminal Service Gateway (TSGW). Com esta última é possível configurar o cliente da sessão de terminal (Remote Desktop Client - RDC) para se ligar ao servidor, não recorrendo unicamente ao protocolo RDP, mas sim através de RDP sobre HTTPs. Ou seja, antes da sessão de RDP propriamente dita ser estabelecida, já terá de estar estabelecido um túnel HTTP com HTTP *Secure Sockets Layer/Transport Layer Security (SSL/TLS)* até ao TSGW, conforme pode ser constatado na figura 4.2. Isto permite que as vulnerabilidades existentes numa aplicação deixem de ser exploráveis no troço de rede entre o cliente RDC e o TSGW. Se houver a garantia de que o troço restante, entre o TSGW e o TS, não está acessível por alguém mal intencionado, poderemos passar a usufruir da mesma aplicação sem qualquer tipo de alteração, mas já de uma forma segura.

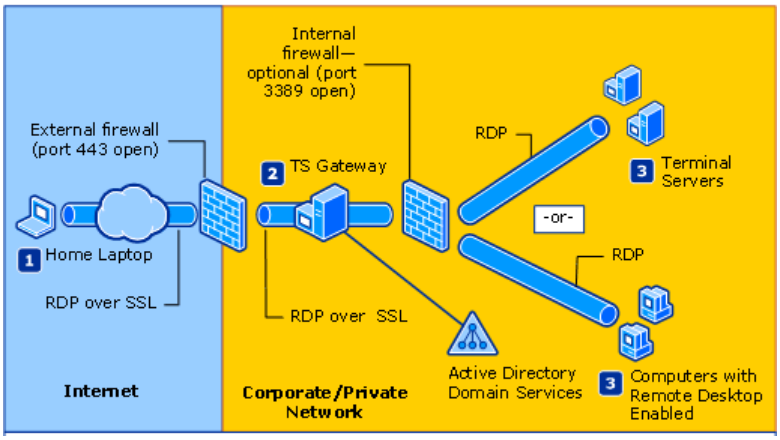


Figura 4.2: Terminal Services Gateway associado ao RemoteApp[9]

Tal como no estabelecimento da sessão de terminal, o utilizador tem de apresentar algum tipo de credenciais para estabelecer o canal de comunicação com o TSGW, nomeadamente *login/password* ou *smartcard*, no entanto, seria necessário proceder a duas autenticações para a execução de uma aplicação remota, para além da já anteriormente executada no acesso ao computador. Para eliminar estas duas adicionais, é possível configurar o acesso do cliente RDC para que o mesmo usufrua de SSO, eliminando o trabalho adicional do utilizador.

Outro sistema utilizável seria o X Window Protocol encapsulado através de um túnel

SSH. Isto permitiria colmatar a falta de segurança do primeiro através do túnel cifrado do segundo, passando assim a execução remota das aplicações a ser virtualmente segura. Apesar da existência desta alternativa ela não se apresenta como a mais interessante para a UA, dado que a maior parte das aplicações candidatas a serem executadas remotamente são aplicações Windows.

4.9 certificados digitais para utilizadores

Como referido na subsecção Autenticação PKI da secção 3.3 é possível atribuir a cada utilizador um certificado digital que ele poderá apresentar sempre que quiser provar digitalmente a sua identidade. Como o certificado reside fisicamente num dispositivo pode ser conjugado com credenciais do tipo login/password para obtenção de autenticação forte. Recomenda-se assim que para acesso a sistemas considerados mais críticos, nomeadamente aplicações de serviços académicos, serviços financeiros entre outras, seja implementada autenticação forte com base em certificados digitais.

4.9.1 *Smart-cards*

Smart-card é um dos formatos possíveis em que um certificado digital de um cliente pode ser armazenado. Basicamente tem a forma de um cartão de crédito, sendo por vezes acumulada a função de *smart card* com outras, nomeadamente identificação, cartão de proximidade (RFID), entre outras. Tem uma desvantagem que se prende com a necessidade de existência de leitores de *smart-cards* ligados aos computadores em que se pretenda usá-los.

Já na secção 3.3 o Cartão de Cidadão foi referido como sendo um caso particular dos *smart-cards*. No caso de acesso a recursos (informáticos ou não), como serviços de atendimento, submissão de informação considerada crítica, entre outros casos, talvez fosse de aproveitar a existência deste mesmo cartão para garantir uma segurança acrescida nas operações efectuadas.

A UA, há já alguns anos, utiliza cartões emitidos por uma instituição bancária para identificação dos seus funcionários e alunos. Mais recentemente estes passaram a incluir no seu interior um RFID. Apesar disto, a sua potencialidade não era explorada pois já anteriormente tinham começado a ser utilizados outros cartões de RFID, por exemplo para acesso a parques, edifícios, etc. No âmbito do Programa "Eficiência Energética na UA", está actualmente em fase de implementação o "Cartão Único". Com isto pretende-se uniformizar os cartões de acesso aos mais variados edifícios/equipamentos, passando a ser utilizado unicamente o cartão de identificação que cada pessoa já possui. Para além de deixar de haver a necessidade da emissão/transporte de outros cartões, com este projecto pretende-se também obter reduções nos consumos energéticos na UA, através da detecção e controlo do número de pessoas no interior dos edifícios (o que permitirá a desactivação automática da alimentação eléctrica de equipamentos não essenciais, nomeadamente os de climatização, iluminação, entre outros).

Por outro lado, ao nível da segurança de pessoas, em situações de emergência pretende-se que venha a permitir a indicação das pessoas que se encontram dentro de cada edifício em cada momento, e na perspectiva dos bens imóveis, disponibilizar meios que possibilitam aferir o estado de abertura (ou fecho) das portas de acesso aos edifícios.

Este cartão poderá também ter utilidade em Sistemas Informáticos, nomeadamente no acesso a equipamentos, aplicações, ou dados considerados mais críticos, em que a utilização

da autenticação via *login/password* poderia ser complementada com a leitura do RFID do cartão do utilizador, procedendo-se assim a autenticação forte.

4.10 *One Time Passwords – OTP*

A utilização de credenciais baseadas em password tem como principal problema a gestão da sua política. É difícil encontrar o equilíbrio entre a sua complexidade, a sua dimensão, o seu período de validade entre outras características da password e a necessidade que o utilizador vai ter de tomar nota da mesma, porque ela é demasiado difícil de decorar. Se o utilizador sentir essa necessidade todo o objectivo de uma boa política de password é destruído. Uma das formas encontradas de contornar esta limitação é fornecer ao utilizador um dispositivo que gera passwords temporárias. Tipicamente tem a dimensão de um porta-chaves e inclui um mostrador LCD onde é mostrada a password que deverá ser usada na próxima autenticação. O método de geração das passwords recorre à hora a que a password é gerada ou ao número de password geradas até então. A isto pode ser associada a utilização de um PIN ou password simples que o utilizador conheça, passando assim a ter-se autenticação forte. Não há qualquer tipo de garantia de que a password não possa ser capturada, mas dado que a sua utilização válida é uma e apenas uma, de nada servirá a um terceiro o seu conhecimento se já não a puder utilizar.

A título de exemplo podem ser referidos como destinos potenciais, os sítios web/aplicações que seja desejável possuírem uma autenticação mais forte do que o recurso ao simples par *login/password*, mas cuja integração com leitores de cartões, sensores biométricos, entre outros, seja difícil, ou em que os pontos de utilização seja de tal ordem diversa, que os custos de aquisição dos mesmos sejam uma limitação. Este sistema poderia eventualmente ser utilizado conjuntamente com o envio de SMS, ou seja, a *password* seria enviada para o telemóvel do destinatário da mesma. Assim só poderia ser utilizada pelo portador do mesmo, virtualmente o seu dono.

4.11 Os Pólos da UA

A Universidade de Aveiro encontra-se sediada no Campus de Santiago, no entanto possui vários pólos que distam algumas dezenas de quilómetros deste. Nestes pólos são utilizados alguns recursos informáticos que residem localmente e outros que residem no Campus de Santiago. Isto é possível através de ligações de rede que vão desde feixes de rádio frequência, a ligações dedicadas em fibras óptica. No entanto qualquer uma delas é passível de falha, pelo que é desejável que sejam identificados e disponibilizados os serviços críticos mínimos para que durante uma falha ao menos os recursos existentes localmente continuem utilizáveis.

Identificação dos serviços críticos

Os serviços que são considerados mínimos ao funcionamento dos computadores clientes, aos servidores, aos equipamentos de rede, entre outros, são:

- DNS
- DHCP
- AD

Dos serviços enumerados atrás, aquele que é passível de ser alvo de interesse deste trabalho é o *Active Directory*. Este serviço, como muitos outros, possui fragilidades ao nível físico, ou seja, se houver acesso ao servidor que disponibiliza o serviços por pessoas mal intencionadas, poderá ser comprometida a informação nele residente, nomeadamente as cifras das *passwords* dos utilizadores. Dado que as condições físicas de alojamento não são tão seguras nem controladas como as existentes no *datacenter* da UA, a probabilidade deste comprometimento ocorrer é bastante superior e eventualmente pior do que isto é, no caso de acontecer, isso não ser detectado para poderem ser tomadas medidas de controlo de danos.

Até à versão anterior da *Active Directory* (2003), a existência de um KDC implicava uma réplica total da informação constante no directório. Com a versão 2008 passou a ser possível a existência de um KDC que apenas possui os dados referentes aos objectos (contas de utilizadores, grupos, etc) que sejam configurados explicitamente para serem replicados para esse servidor. Assim, por suposição no pólo X, o KDC lá existente apenas deverá conter a informação relativa aos 300 utilizadores que habitualmente nele se encontram (alunos, professores, administrativos) e não as dezenas de milhares existentes na AD. A este tipo de KDC a Microsoft designou de *Read Only Domain Controller* (RODC).

Torna-se assim necessária a substituição dos servidores existentes ou a instalação de novos nos vários pólos da UA, garantindo que é possível, em caso de falha da ligação remota continuar a realizar a autenticação dos utilizadores localmente, sem que isto crie riscos para toda a infraestrutura.

Capítulo 5

Conclusões e trabalho futuro

Tal como era desejável e previsível, o estudo que a presente dissertação implicou veio permitir um aprofundar do conhecimento na área da autenticação nos sistemas informáticos, de uma forma genérica e em particular dos existentes na Universidade de Aveiro (UA).

Como metodologia seguiu-se uma abordagem do mais lato para o mais específico. Começou-se por estudar a problemática da Segurança da Informação (digital ou outra). Em seguida enveredou-se pela análise dos riscos a que essa informação está sujeita, bem como dos mecanismos e tecnologias que permitem detectá-los, minimizá-los e eventualmente mitigá-los. Posteriormente foram efectuados estudos mais específicos, já no âmbito da autenticação e dos seus sistemas, tecnologias, mecanismos e protocolos. Por fim foi efectuado um levantamento e análise do caso particular da autenticação na Universidade de Aveiro, no qual foram identificados sistemas, aplicações e serviços cuja utilização fossem de âmbito alargado para os vários tipos de utilizadores existentes (alunos, funcionários docentes e funcionários não docentes). Após este levantamento foram indicadas algumas propostas de melhoria relativas aos sistemas de autenticação propriamente ditos, aos seus métodos, protocolos, comunicações ou mecanismos.

Não poderia deixar de ser realçado que, tal como é explicitado na presente dissertação, a garantia de protecção da autenticação não reside apenas nos protocolos, mecanismos, ou tecnologias existentes, mas também numa série de outros factores que vão desde a infraestrutura física à componente humana. Por muito que os primeiros sejam seguros, caso os comportamentos dos utilizadores não o sejam todo o processo de autenticação será posto em causa.

Tem igualmente que ser destacado que, tal como diz Mark Stamp, "In general there is no best authentication protocol. What is best for a particular situation will depend on many factors" [21]. Ou seja, não existe um único protocolo. Coexistem vários com vantagens e desvantagens, e em função do cenário de utilização deve-se optar pelo mais adequado.

Após o levantamento dos principais sistemas de autenticação em uso na UA são indicadas algumas propostas de melhoria.[1] Umas poderão ter um custo elevado de adopção, mas já outras terão com certeza custos significativamente baixos relativamente aos benefícios que proporcionarão nos sistemas, serviços e aplicações. Estes ganhos podem residir quer na vertente de segurança, quer na da usabilidade. A título de curiosidade pode ser indicado que nem sempre para se incrementar a segurança do processo de autenticação é necessário qualquer tipo de alteração do mesmo (ex: TSGW).

Pode ser ainda referido que pelo facto de possuir o papel de administrador de alguns

[1] Por se tratar este de um documento de domínio público, não são referidos casos concretos de falhas.

dos sistemas da UA isso tem vindo a permitir que certas melhorias identificadas fossem, no decurso deste estudo, passando da teoria à prática. Destas gostaria de destacar algumas em que houve envolvimento pessoal, a título exclusivo ou participativo: projecto RCU/Utilizador Universal da UA, alteração da AD de forma a incluir os atributos necessários à autorização dos clientes linux dos laboratórios, recurso ao sistema central de autenticação no acesso aos servidores centrais linux, definição e implementação de políticas de *password* distintas para os diferentes tipos de contas existentes, implementação do *Identity Provider Shibboleth (IdP)* da UA (em federação com FCCN e SAPO), bem como de alguns *Service Providers* (moodle, plataforma de *backoffice* do servidor de sítios institucionais, entre outros), implementação de *Read Only Domain Controllers* num dos pólos da UA.

Trabalho futuro

Apesar de várias propostas terem sido entretanto implementadas algumas surgem na forma de suporte a que outras possam vir a ser encetadas. Por exemplo foi implementado o IdP da UA, mas ele só produz trabalho útil se existirem diversos sítios/aplicações web que usufruam dele. Eventualmente tem de ser efectuado um trabalho de conversão destes últimos, bem como definidas directivas que apontem no sentido da sua utilização no caso do desenvolvimento de produtos novos.

Para uma mais completa integração dos clientes linux existentes nos laboratórios e gabinetes deveria ser efectuado uma análise da viabilidade da disponibilização do acesso ao conteúdo do sistema de armazenamento de dados central (ARCA), através de protocolo NFSv4, recorrendo-se assim a Kerberos, bem como da disponibilização do acesso ao serviço de impressão com base no IPP. Outro trabalho aparentemente interessante reside na autorização de elevação de privilégio a root destes clientes (*sudoers*) com base num grupo do LDAP/AD. Actualmente isso só é possível com recurso a contas locais.

Actualmente para o acesso a determinados serviços por parte de terceiros (externos à UA) é necessário o recurso a uma conta de visitante (ex: costa@visit.uaveiro.eu). Para a sua obtenção é necessário um registo, passando o utilizador em causa a ter de gerir mais uma identidade electrónica. Talvez fosse de considerar o recurso a autenticação baseada em OpenID para este género de acessos.

Apêndice A

Generic Security Service Application Program Interface (GSSAPI)

Como o próprio nome indica, *Generic Security Service Application Program Interface[22] (GSSAPI)* permite que os programadores escrevam aplicações que são independentes no que diz respeito à segurança. Isto é, não têm de se preocupar em implementar para um determinado tipo de plataforma, mecanismo de segurança, tipo de protecção ou protocolo de transporte. Apesar de as aplicações poderem ter controlo sobre determinados aspectos de segurança, o programador consegue escrever um programa que é ignorante na forma como deve proteger os dados que irão fluir na rede. Deste modo, um programa que recorra às potencialidades do GSSAPI possui bastante portabilidade no que respeita à segurança de rede. Esta é mais que todas as outras a principal característica do GSSAPI.

O GSSAPI não disponibiliza por si só serviços de segurança. Ao contrário, é sim uma *framework* que fornece estes a quem a evoca (callers), e que é suportada por um conjunto de mecanismos e tecnologias como Kerberos, chaves públicas, etc, conforme é mostrado na figura que se segue:

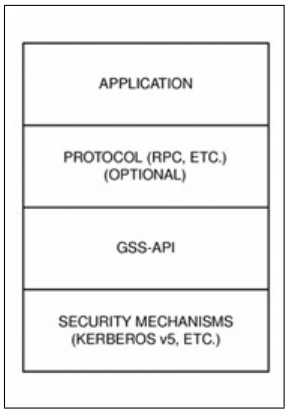


Figura A.1: GSSAPI *Framework*

Basicamente o GSSAPI consiste em duas coisas:

1. Criação de um contexto de segurança em que informação pode ser trocada entre duas aplicações. Este contexto permite um conhecimento bilateral entre as aplicações e a troca de dados entre elas, enquanto o mesmo durar.

2. Uso de um ou mais tipos de protecção, conhecidos por *security services*, na informação a ser transmitida.

Claro que o GSSAPI é mais complexo que isto. Para além do enunciado anteriormente, o GSSAPI incluí: conversão de dados, verificação de erros; delegação de privilégios de utilizador; visualização de informação e comparação de identidades. O GSSAPI inclui assim um múltiplo número de funções que permitem o estabelecimento e a exploração deste contexto pelas aplicações, conforme figura abaixo:

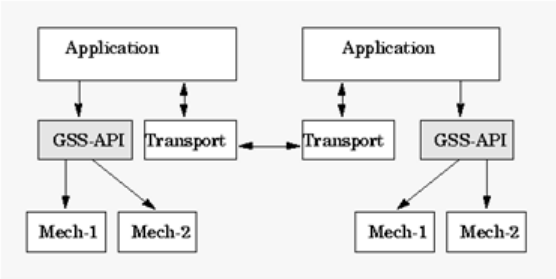


Figura A.2: GSSAPI Cliente/Servidor

A.1 Portabilidade Aplicacional

Como foi referido anteriormente o GSSAPI disponibiliza vários tipos de portabilidade às aplicações:

Mechanism independence: o GSSAPI possui uma interface genérica para os mecanismos para os quais foi desenvolvida. Uma aplicação ao especificar um mecanismo por omissão, não precisa de conhecer o mecanismo em si (por exemplo Kerberos), ou mesmo quais os tipos de mecanismos que vão ser usados. Ou seja, quando uma aplicação encaminha uma credencial de utilizador para um servidor, não precisa de conhecer o formato da credencial, ou de algum mecanismo da aplicação, nem se as credenciais são armazenadas de alguma forma e onde, para futuro utilização pela aplicação

Protocol independence: O GSSAPI é independente do tipo de protocolo de comunicação. Tanto pode ser usado em aplicações que usam sockets, RCP, TCP/IP, etc.

Platform independence: O GSSAPI é completamente indiferente ao tipo de sistema operativo em que a aplicação está a ser executada.

Quality of Protection independence: *Quality of Protection (QOP)* é o nome dado ao tipo de algoritmo usado na encriptação de dados, ou na criação das *cryptographic tags*; o GSSAPI permite assim ao programador ignorar o QOP, e utilizar o que está definido por omissão para ser usado pelo GSS-API. Se desejar pode especificar o QOP que pretende utilizar.

A.2 Serviços de Segurança

O serviço de segurança mais básico que é disponibilizado pelo GSSAPI é o de autenticação. Para além deste são disponibilizados mais dois serviços, que são dependentes dos mecanismos existentes:

Integridade: Nem sempre é suficiente saber que a aplicação que nos está a enviar dados é quem diz que é. Os dados propriamente ditos podem ser corrompidos ou comprometidos. O GSS-API providencia que juntamente com os dados vá uma *cryptographic tag*, conhecida por *Message Integrity Code (MIC)*, e que vai permitir que os dados antes de usados pela aplicação sejam verificados, de modo a garantir que são os que efectivamente foram enviados. A integridade da informação é assim garantida.

Confidencialidade: Tanto a autenticação como a integridade não invalidam que a informação trocada não seja de algum modo interceptado e que terceiros a possam ler. O GSSAPI permite assim que a informação seja encriptada, caso os mecanismos existentes o suportem.

A.3 Mecanismos

A implementação do GSS-API trabalha com diversos mecanismos sendo o mais vulgar o mecanismo de segurança Kerberos v5.

Apêndice B

Shibboleth na UA – Novo *Service Provider (SP)*

B.1 Instalação Linux

A instalação poderá ser efectuada de três métodos distintos:

pacotes RPM:

(caso existam para o Sistema Operativo ou distribuição em causa)

(`https://spaces.internet2.edu/display/SHIB2/NativeSPLinuxSRPMBuild`)

(verificar os Sistemas Operativos e as distribuições em `http://shibboleth.internet2.edu/shib-which-version.html`)

- Efectuar o download de todos os pacotes existentes
  (`http://shibboleth.internet2.edu/downloads/shibboleth/cppsp/2.1/RPMS/`)
  Exemplo para Red Hat Enterprise 5:

```
curl –O http://shibboleth.internet2.edu/downloads/shibboleth/
cppsp/2.1/RPMS/i386/RHE/5/log4shib-1.0-1.i386.rpm \
     –O http://shibboleth.internet2.edu/downloads/shibboleth/
cppsp/2.1/RPMS/i386/RHE/5/xerces-c-2.8.0-1.i386.rpm \
     –O http://shibboleth.internet2.edu/downloads/shibboleth/
cppsp/2.1/RPMS/i386/RHE/5/xml-security-c-1.4.0-1.i386.rpm \
     –O http://shibboleth.internet2.edu/downloads/shibboleth/
cppsp/2.1/RPMS/i386/RHE/5/xmltooling-1.1-1.i386.rpm \
     –O http://shibboleth.internet2.edu/downloads/shibboleth/
cppsp/2.1/RPMS/i386/RHE/5/opensaml-2.1-1.i386.rpm \
     –O http://shibboleth.internet2.edu/downloads/shibboleth/
cppsp/2.1/RPMS/i386/RHE/5/shibboleth-2.1-1.i386.rpm
```

- Instalar os pacotes
  Exemplo para Red Hat Enterprise 5:

```
rpm -ivh log4shib-1.0-1.i386.rpm \
    xerces-c-2.8.0-1.i386.rpm \
```

```
xml-security-c-1.4.0-1.i386.rpm \
xmltooling-1.1-1.i386.rpm \
opensaml-2.1-1.i386.rpm \
shibboleth-2.1-1.i386.rpm
```

Durante a instalação, os vários componentes do Shibboleth serão colocados nos directórios *default* da distribuição, tipicamente:

- Os ficheiros de configuração do Shibboleth estarão localizados em `/etc/shibboleth/` e a configuração Apache em `/etc/httpd/conf.d/shib.conf`
- shibd será instalado em `/usr/sbin` e poderá ser gerido usando `/sbin/service` e `/sbin/chkconfig`
- A versão apropriada de mod_shib e de outros módulos estarão em `/usr/lib/shibboleth/`
- Os Logs residirão em `/var/log/httpd/native.log` e `/var/log/shibboleth/shibd.log`

Recompilar com base em pacotes SRPM:

(caso existam para o Sistema Operativo ou distribuição em causa)

(o *download* dos pacotes pode ser feito em `https://spaces.internet2.edu/display/SHIB2/NativeSPLinuxSRPMBuild`)

Este método é muito semelhante ao método seguinte, mas bastante mais automatizado, logo mais fácil para pessoas menos experientes. O conjunto de pacotes resultante poderá ser usado para instalar mais facilmente em máquinas idênticas. O processo basicamente consiste em:

- Efectuar o download dos pacotes SRPM existentes (`http://shibboleth.internet2.edu/downloads/shibboleth/cppsp/latest/SRPMS/`)
- csecurity-c, xml-security-c, xmltooling, opensaml, shibboleth
  Exemplo:

  ```
  rpmbuild --rebuild package.src.rpm
  rpm -ivh /usr/src/redhat/RPMS/i386/package-version-rec.arch.rpm
  rpm -ivh /usr/src/redhat/RPMS/i386/package-devel-version-rec
  .arch.rpm
  ```

Para mais informações sobre como recompilar, consultar: `http://bradthemad.org/tech/notes/patching/rpms.php`. Os componentes, bem como os ficheiros de logs encontram-se nas localizações já descritas no método anterior.

Compilar com base nas *sources*:

(o *download* das *sources* pode ser feito em `https://spaces.internet2.edu/display/SHIB2/NativeSPLinuxSourceBuild`)

- Efectuar o download das sources (`http://shibboleth.internet2.edu/downloads`, etc)
- Compilar as dependências e o Shibboleth 2.X SP propriamente dito

- definir a variável de ambiente LD_LIBRARY_PATH com a localização das dependências e as bibliotecas do Shibboleth (exemplo: export LD_LIBRARY_PATH = /opt/shibboleth-sp/lib)

Neste tipo de instalação, os ficheiros de log, by default, ficam numa localização diferente: /opt/shibboleth-sp/var/log.

Independentemente do método de instalação usado, são agora necessárias algumas configurações no que ao servidor Apache diz respeito, para isso basta:

1. alterar o ficheiro httpd.conf para:
   - inserir no VirtualHost respectivo a seguinte directiva:
     Include /opt/shibboleth−sp/etc/shibboleth/apache22.config.
   - Modificar a directiva UseCanonicalName para On
   - Verificar se a directiva ServerName está definida convenientemente e que o Apache tem SSL activo
2. Reiniciar o Apache
3. Por fim executar o serviço do Shibboleth SP. Consoante a instalação será `/usr/sbin/shibd&`, ou `/opt/shibboleth-sp/sbin/shibd&`, ou semelhante, dependendo da localização dos componentes.

Para testar a operacionalidade da instalação do Shibboleth basta aceder na própria máquina a `https://localhost/Shibboleth.sso/Status`. Se tudo estiver bem, deverá ser devolvido um "OK".

B.2 Instalação Windows

Existem várias configurações possíveis de implementar. A mais vulgar no Sistema Operativo em causa é a que recorre ao IIS como servidor Web. No entanto outras são possíveis, nomeadamente usar um servidor web Apache/Tomcat. Nesse caso o processo de configuração do servidor Web seria semelhante ao descrito anteriormente para distribuições Linux, com as necessárias adaptações.

Em primeiro lugar é necessário proceder ao download do pacote do Shibboleth SP em `http://shibboleth.internet2.edu/downloads/shibboleth/cppsp/latest/win32`.

Posteriormente pode ser executado o instalador (*.msi). Durante a execução do mesmo, serão solicitadas algumas informações. Salvo alguma necessidade especial, os valores para as mesmas deverão ser as propostas by default. Durante a instalação são efectuadas algumas configurações do Windows e do IIS, de variáveis de sistema, e é inserido um novo serviço. Para isto decorrer com normalidade também no caso da presença de um IIS7, é necessária a prévia instalação da feature role IIS 6 Management Compatibility.

De seguida, para tomarem efeito as alterações encetadas pelo instalador é necessária a realização de um reboot ao servidor.

Para a verificação da instalação, basta:

1. IIS 5 e IIS 6

- Executar o MMC do IIS Manager. Posteriormente abrindo as propriedades do servidor em causa, seleccionar o tabulador ISAPI Filters, e adicionar um novo filtro com a designação Shibboleth e cuja biblioteca é o ficheiro isapi_shib.dll existente na pasta lib
  shibboleth da instalação do Shibboleth. A prioridade deve ser High, e uma vez carregado o filtro o mesmo deve surgir imediatamente a seguir a sspifilt. Reiniciar o serviço de IIS e posteriormente verificar se uma seta verde surge junto ao filtro do Shibboleth e se no Event Viewer e nos Logs de Shibboleth tudo arranca com sucesso.
- Ainda nas propriedades do servidor, mas na tabulador "Home Directory", carregar no botão "Configuration", seguido da adição de um novo mapeamento, para a extensão .sso e cujo executável será o ficheiro isapi_shib.dll. A opção "limit verbs"e "Check that file exists"devem ser desactivadas.
- Reiniciar o serviço de IIS.

2. IIS 7

- Executar o MMC do IIS Manager. Aceder ao servidor em causa e na área à direita aceder a ISAPI Filters. Adicionar um novo filtro com a designação Shibboleth e cuja biblioteca é o ficheiro isapi_shib.dll existente na pasta lib
  shibboleth da instalação do Shibboleth.
- Novamente acedendo ao servidor, na área da direita aceder a Handler Mappings. Na lista de acções executar Add Module Mapping, indicando como extensão .sso, como Módulo IsapiModule, e como executável o ficheiro isapi_shib.dll. Premir o botão Request Restrictions..., no tabulador Mapping activar Invoke... e seleccionar File. No tabulador Verbs seleccionar All verbs e por fim no tabulador Access seleccionar Execute. Fechar todas as janelas indicando OK. Deverá surgir uma questão sobre se pretende adicionar uma excepção para a extensão em causa no item ISAPI and CGI Restrictions. A resposta deverá ser afirmativa.
- Reiniciar o serviço de IIS.

B.3 Configuração

A configuração de um Shibboleth SP é independente do Sistema Operativo e do servidor Web, pelo que a partir de agora todas as informações mencionadas se aplicam a qualquer um dos sistemas e configurações abordadas anteriormente.

As configurações gerais são efectuadas no ficheiro shibboleth2.xml, que deverá estar localizado em `/etc/shibboleth` ou `opt/shibboleth-sp/etc/shibboleth`. Neste ficheiro tem de ser indicado:

- O hostname e ID da instância de IIS, através da directiva:

  <Site id="ID_do_site" name="host.name.fqdn"/>.

- O(s) caminho(s) cujo acesso estará(ão) protegido(s) por uma sessão Shibboleth, através de directivas que residirão dentro da directiva ¡RequestMap applicationId="default≫:

```
<Host name="host.name.fqdn">
<Path name="localizacao/SeguraComShibboleth" authType="shibboleth"
requireSession="true" />
</Host>
```

- A designação da instância de Shibboleth e do URL para onde será redireccionado o *browser* caso no acesso ao site não tenha sido especificado um URL apropriado:

```
entityID="https://host.name.fqdn/shibboleth"
homeURL="https://host.name.fqdn/urlredirection"
```

- o IdP responsável pelo método de *login*, que se encontra dentro da directiva iniciada por <SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="Intranet":

```
relayState="cookie" entityID="https://idp.ua.pt/idp/shibboleth">
```

- A localização da Metadata do IdP, através da directiva:

```
<MetadataProvider type="XML" uri="http://idp.ua.pt/idp/idp-metadata
.xml" backingFilePath="idp.ua.pt-metadata.xml">
<!-- <SignatureMetadataFilter certificate="idp.pem" /> -->
</MetadataProvider>
```

Após terminar a edição do ficheiro shibboleth2.xml, poder-se-á efectuar um restart ao serviço do Shibboleth e verificar se as novas definições são carregadas sem problemas. Para isso devem ser consultados os logs do Shibboleth durante o processo de start do serviço e quando um *browser* aceder a uma das localizações protegidas por sessão Shibboleth. Estando tudo a operar convenientemente o browser deverá ser redireccionado para a página do IdP da Universidade de Aveiro (`https://idp.ua.pt/idp/Authn/UserPassword`), onde o utilizador inserirá as credenciais e será com estas autenticado. Se todo o processo de autenticação decorrer normalmente, o browser deverá mais uma vez ser redireccionado, desta vez, para a página que inicialmente o utilizador pretendia aceder, já com a sessão Shibboleth estabelecida.

Para se confirmar o estado da Sessão Shibboleth poderá ser consultado o seguinte endereço: `http://host.name.fqdn/Shibboleth.sso/Session`. Entre outras informações (Miscellaneous) será indicado o número de atributos que foram libertados pelo IdP, e mapeados e filtrados pelo SP para sua utilização.

O mapeamento consiste basicamente nisso mesmo, no mapeamento de atributos libertados pelo IdP em atributos a utilizar pelo Service Provider. Assim a designação de um atributo que faz sentido no contexto do IdP pode ser convertida noutra que faça mais sentido no contexto do SP. Por exemplo o atributo n_mecanografico é mapeado em n_aluno. O ficheiro onde fica definido os mapeamentos é o attribute-map.xml existente na mesma localização que o shibboleth2.xml. Quase no final do ficheiro está a seguinte directiva:

```
<Attribute name="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
id="persistent-id">
<AttributeDecoder xsi:type="NameIDAttributeDecoder" formatter=
"$NameQualifier!$SPNameQualifier!$Name" />
</Attribute>
```

Após isto poderão ser inseridas entradas com mapeamentos do tipo:

```
<Attribute name="designação no IdP" id="designação no SP" />
```

Para além do mapeamento anteriormente descrito poderá ser efectuado algum tipo de processamento ao conteúdo dos atributos recebidos para ser gerado um outro ou alterado o próprio conteúdo. Por exemplo o atributo Unidade pode ser preenchido com DETI no caso do atributo n_mecanografico ter valores entre 30000 e 35000. Por omissão de configuração topos os atributos são permitidos e permanecem inalterados. O ficheiro onde fica definida a política de filtragem é o attribute-policy.xml existente na mesma localização que o shibboleth2.xml.

Bibliografia

[1] The Institute of Internal Auditors. The Institute of Internal Auditors (IIA) Web Site. `http://www.theiia.org/`, Dezembro 2009.

[2] Donn B. Parker. *Fighting Computer Crime*. John Wiley & comercial Sons, Inc, New York, 1998.

[3] CyberSecureOnline.com, LLC. `http://www.cybersecureonline.com/`, Dezembro 2009.

[4] SCP Corporate. *SCNA: Network Defense and Countermeasures (study guide)*. Element K Courseware, New York, 2008.

[5] André Zúquete. *Segurança em redes informáticas, 2ª edição aumentada*. FCA - Editora de Informática, Lisboa, 2007.

[6] Verisign Web Site. `http://www.verisign.com`, Dezembro 2009.

[7] William Stallings. *Cryptography and Network Security - Principles and Practice (2nd edition)*. Prentice Hall, New Jersey, 1999.

[8] Matt Bishop. *Computer Security - Art and Science*. Addison-Wesley Professional, New Jersey, 2002.

[9] Microsoft Corporation. Microsoft TechNet. `http://technet.microsoft.com/`, Março 2009.

[10] Charlie Kaufman, Radia Perlman, and Mike Speciner. *Network Security - Private Communication in a public world (2nd edition)*. Prentice Hall PTR, New Jersey, 2002.

[11] Dieter GollMann. *Computer Security (2nd edition)*. John Wiley & comercial Sons, Ltd, West Sussex, 2005.

[12] Kerberos Team. Mit kerberos site. `http://web.mit.edu/kerberos/www`, Março 2009.

[13] Kerberos Consortium. MIT Kerberos Consortium. `http://www.kerberos.org/`, Março 2009.

[14] USC/ISI. USC/ISI Kerberos Page. `http://www.kerberos.info/`, Março 2009.

[15] Network Working Group. RFC 1510 - The Kerberos Network Authentication System Service (v5). `http://www.ietf.org/rfc/rfc1510.txt`, September 1993.

[16] OpenID Foundation. OpenID. `http://openid.net/`, Abril 2009.

[17] Internet 2. Shibboleth. http://shibboleth.internet2.edu/, Maio 2009.

[18] Carlos Costa. Discovery Service da UA. http://idp.ua.pt/discovery, Janeiro 2009.

[19] Carlos Costa. Identity Provider da UA. https://idp.ua.pt/idp, Janeiro 2009.

[20] William Stallings. *Network Security Essentials - Applications and Standards (2nd edition).* Prentice Hall, New Jersey, 2003.

[21] Mark Stamp. *Information Security: Principles and Practice.* John Wiley & comercial Sons, Inc, New Jersey, 2005.

[22] The Internet Engineering Task Force (IETF)-Network Working Group. Generic Security Service Application Program Interface-Version 2, Update 1. http://www.ietf.org/rfc/rfc2743.txt, Maio 2009.

[23] SCP Corporate. *SCNA: Advanced Security Implementation (study guide).* Element K Courseware, New York, 2008.

[24] Public-Key Infrastructure Working Group. Public-Key Infrastructure. http://ietf.org/html.charters/pkix-charter.html, Junho 2009.

[25] NIST PKI. NIST PKI Program. http://csrc.nist.gov/pki/, Junho 2009.

[26] Software Engineering Institute (SEI). CERT-Computer Emergency Response Team. http://www.cert.org/, Setembro 2009.

[27] SANS Institute. SANS (SysAdmin, Audit, Network, Security) Web Site. http://www.sans.org/, Setembro 2009.

[28] National Cyber Security Division (NCSD). United States Computer Emergency Readiness Team. http://www.us-cert.gov/, Setembro 2009.

[29] Network Working Group. RFC 1760 - The S/Key One-Time Password System. http://www.ietf.org/rfc/rfc1760.txt, February 1995.